FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Arquitectura distribuída para serviços de rede centralizados

José António Barros Soares

Licenciado em Engenharia Electrotécnica e de Computadores
pela Faculdade de Engenharia da Universidade do Porto

Dissertação submetida para satisfação parcial dos
requisitos de grau de mestre
em
Redes e Serviços de Comunicação

Dissertação realizada sob a supervisão de
Prof. João Isidro Vila Verde,
do Departamento de Engenharia Electrotécnica e de Computadores
da Faculdade de Engenharia da Universidade do Porto

Porto, Dezembro de 2004

Agradecimentos

À minha esposa Cidália, por ter abdicado do marido em prole deste trabalho por incontáveis vezes e por me ter incentivado a nunca desistir, mesmo em momentos que em que tal parecia ser a melhor solução.

À minha mãe Maria da Glória, por me ter ensinado que os melhores caminhos na vida, raramente são os que à partida se afiguram mais fáceis.

Ao meu colega e amigo Ricardo Leite, que esteve sempre disponível para mais um "empurrãozinho", mais uma noitada ou mais um esforço final em busca do objectivo.

Ao Rodrigo Oliveira e ao Victor Calçada, pela preocupação, pelo incentivo e pela ajuda, sempre que foi necessária, ao longo de todo o curso.

Ao Eng.º Pedro Afonso e ao Eng.º Vítor Pinguicha, que apesar de termos sempre duras batalhas empresariais para travar, nunca exigiram que abdicasse deste meu objectivo pessoal em prole das empresas.

Ao Professor Doutor José Manuel Mendonça e ao Eng.º Luís Leal Victor por terem contribuído decisivamente para o "pontapé de saída" deste percurso, que culminou agora nesta dissertação final.

Resumo

O DHCP é um serviço centralizado para gestão automática de endereçamento em redes IP. Este serviço é concebido para operar num ambiente de rede local e deve estar sempre disponível, i.e. a comunicação entre clientes e servidores não deve ser interrompida para não prejudicar o bom funcionamento do serviço.

Devido à flexibilidade de gestão que este serviço potencia, é frequentemente escolhido para operar, não só em redes locais, mas também em redes alargadas constituídas por redes locais distribuídas geograficamente (sites).

Neste contexto, o carácter centralizado deste serviço torna-o vulnerável a falhas de conectividade com redes remotas cujo serviço DHCP seja gerido centralmente. Na ocorrência destas falhas, as máquinas nas redes remotas, ficam impossibilitadas de obter a configuração necessária para ter conectividade local.

Para este problema, propõe-se uma solução de redundância distribuída baseada numa arquitectura de sistemas autónomos dotados de alguma inteligência e distribuídos geograficamente pelos vários locais (sites), que colmate as vulnerabilidades do sistema centralizado (mantendo-se as suas vantagens) e garanta a continuidade do serviço, mesmo nos locais (sites) remotos.

Abstract

DHCP is a centralized network service that provides a framework to automatically allocate network addresses in IP networks.
This service was built to operate in a LAN environment and must be running continuously for correct operation, i.e. servers and clients must be able to exchange DHCP messages all the time.

Due to the management flexibility that this service provides, it is often chosen to operate, not only in a LAN environment, but also in a WAN environment built as a set of distributed LAN's.

Regarding this, the centralized philosophy of the service, makes it vulnerable to connectivity (link) failures with remote sites, becoming impossible to some machines in the remote site, to gain local connectivity with the local network that is being managed by a remote DHCP server.

To overcome this problem, it is proposed a solution of distributed redundancy consisting in an architecture of autonomous systems, with a determined degree of intelligence, each one distributed by the remote sites. It is expected that this solution is able to overcome the vulnerabilities of the centralized system (maintaining the advantages), guaranteeing the continuity of the service even in remote sites.

Índice

Lista de Figuras

Lista de Tabelas

Glossário

ADSL – Asymmetric Digital Subscriber Line
ARP – Address Resolution Protocol
BOOTP – Bootstrap Protocol
CIDR – Classless Inter Domain Routing
DHCP – Dynamic Host Configuration Protocol
ICMP – Internet Connection Messaging Protocol
IP – Internet Protocol
ISP – Internet Service Provider
LAN – Local Area Network
MAC – Media Access Control
PERL – Practical Extraction and Report Language
RFC – Request for Comments
SLA – Service Level Agreement
UDP – User Datagram Protocol
WAN – Wide Area Network

1 Introdução

1.1 Redes públicas e privadas

Quando falamos de redes IP, devemos distinguir entre redes IP públicas e redes IP privadas. As redes IP públicas são aquelas a que nos ligamos para obter conectividade Internet e para as quais precisamos de um, ou vários, endereços IP públicos.
Os endereços IP podem ser configurados manualmente ou automaticamente. Os endereços configurados manualmente são estáticos. Os endereços configurados automaticamente podem ser estáticos ou dinâmicos (ou pseudo-estáticos - caso dos endereços IP fixos fornecidos por um ISP de cabo ou ADSL onde o endereço IP é sempre o mesmo, mas configurado automaticamente).

As redes IP privadas são mantidas dentro de uma organização/empresa/grupo de empresas, recorrendo a endereços IP privados (1) (e.g. 192.168.0.0/16 ou 10.0.0.0/8 CIDR[1] (2)).

A gestão de endereçamento deste tipo de redes é da exclusiva responsabilidade do departamento de informática/comunicações das respectivas empresas.

Numa rede privada, tal como numa rede pública, a atribuição de endereços aos equipamentos ligados à rede, pode ser feita, manualmente ou automaticamente (3).

A primeira solução (configuração manual) não é adequada para redes distribuídas por diversos locais físicos ou mesmo para campus informáticos porque implicaria uma configuração manual por cada vez que surgisse uma máquina nova na rede ou de cada vez que uma máquina se deslocasse entre sub-redes. Isto dificultaria a gestão da rede além de representar um custo elevado em recursos humanos afectos a esta tarefa.

A segunda solução recorre a um serviço de rede designado por DHCP (Dynamic Host Configuration Protocol). Este serviço permite a configuração automática de endereçamento numa determinada sub-rede IP.
Esta solução funciona bem em campus informáticos (entenda-se que pode existir mais do que uma rede lógica, mas todas estão fisicamente próximas), mas tem inconvenientes em redes distribuídas por vários locais (fisicamente distantes). Estes inconvenientes estão associados à necessidade do serviço DHCP estar sempre disponível no caso da configuração automática. Num cenário de indisponibilidade, a configuração irá falhar, o que é mais provável de acontecer em redes geograficamente afastadas do que em campus informáticos.

[1] Classless Inter Domain Routing

São vários os serviços que, por questões de flexibilidade de gestão requerem que a mesma seja centralizada (e.g. DHCP, DNS[2]).

Por considerar que o serviço DHCP é o que tem o papel mais crítico (por depender dele uma correcta ligação à rede) e por ser o que menos soluções de redundância disponibiliza (ver secção 3.2), foi o escolhido para ser alvo de análise e melhoria.

A análise do conceito especificamente para o serviço DHCP não significa que não seja válido para outros serviços. Num âmbito mais alargado, perspectiva-se a estruturação de um modelo genérico mais abrangente. No âmbito desta dissertação, no entanto, o objectivo é de dedicação exclusiva ao serviço DHCP, não extrapolando para outros exemplos.

1.2 Constrangimentos do serviço DHCP

O serviço DHCP apresenta alguns constrangimentos.

O primeiro constrangimento está relacionado com a necessidade dos clientes do serviço periodicamente contactarem o servidor. Isto implica que, um determinado caminho de rede (aquele que permite o acesso ao servidor) se torne crítico, estando obrigado a uma fiabilidade máxima. Isto é verdade porque na ausência de serviço DHCP os clientes do serviço perdem acesso à rede, inclusive o acesso à rede local (e.g. assumindo que as impressoras são impressoras de rede, ficam impossibilitados de imprimir,).

O segundo constrangimento está relacionado com uma das suas vantagens: a flexibilidade proporcionada pela gestão centralizada. Como não queremos abdicar desta vantagem, somos forçados a utilizar um só servidor central independentemente da dimensão da rede[3]. Por esta razão, no cenário de uma rede alargada de redes locais distribuídas geograficamente, não podemos usar mais servidores DHCP, pois perderíamos toda a flexibilidade de gerir centralmente o serviço.

São estes constrangimentos, que pretendemos ultrapassar no âmbito desta dissertação, propondo uma solução de redundância distribuída.

1.3 Redundância distribuída

Foi escolhido este título para a arquitectura que se discute nesta dissertação, por se tratar de um tipo de redundância que não pretende ser uma simples duplicação do objecto.

No contexto da engenharia, entende-se como redundância de um determinado sistema ou objecto, a colocação de um sistema ou objecto exactamente iguais aos iniciais para que, em caso de falha, estes sejam substituídos (4).

[2] Domain Name System

[3] O protocolo prevê a existência do servidor secundário, mas sempre com o objectivo de criar uma redundância centralizada

Com a redundância distribuída, os sistemas redundantes não desempenham um papel semelhante, passivo, relativamente aos sistemas principais. Desempenham sim, um papel activo e dotado de inteligência, mesmo na ausência de falhas.

Pretende-se uma arquitectura de agentes autónomos, capazes de proporcionar um cenário de imunidade a falhas, quer aos clientes, quer ao servidor.

1.4 Objectivos

Nesta dissertação começa-se por abordar de forma sucinta e detalhada o protocolo DHCP. Torna-se importante perceber bem o funcionamento, quer do lado do cliente, quer do lado do servidor para entender onde é que os agentes distribuídos actuam os seus algoritmos (recorde-se que se pretendem agentes dotados de inteligência).

Discutido o protocolo em detalhe, abordam-se os vários cenários da sua utilização e avaliam-se as vantagens e desvantagens de cada um.

É feita uma análise de fiabilidade aos vários cenários mais usados nos dias de hoje. Estuda-se as várias soluções existentes de redundância, para um melhor enquadramento da arquitectura proposta para ultrapassar os constrangimentos.

Posteriormente, aborda-se o conceito, propõe-se objectivos a cumprir e propõe-se uma solução, a qual foi objecto de uma implementação para demonstração do conceito e exequibilidade do mesmo. Esta proposta é alvo de análise detalhada, para demonstrar as potencialidades e levantar constrangimentos que surgem ao cumprimento dos objectivos propostos.

1.5 Organização

No capítulo 2 começamos por descrever, numa primeira fase, de forma sucinta os objectivos, o funcionamento e a implementação do serviço DHCP. Numa segunda fase, faz-se uma descrição mais detalhada do protocolo, com especial ênfase nos diferentes estados do serviço.

Conhecido em detalhe o serviço, o capítulo 3 aborda os diferentes cenários de utilização do mesmo. Estes cenários são analisados do ponto de vista da sua fiabilidade e são analisadas algumas propostas já existentes para o aumento da mesma, nas diversas situações de utilização.

Evidenciados os constrangimentos e/ou vulnerabilidades do serviço, no capítulo 4, descreve-se a proposta de solução: a redundância distribuída de gestão centralizada.
Ilustra-se a arquitectura da proposta de solução do ponto de vista da localização na rede, modo de funcionamento e estratégias de implementação. Descreve-se os vários algoritmos utilizados na implementação referindo

vantagens e desvantagens de cada um e os constrangimentos encontrados. São propostas alternativas e testam-se algumas dessas propostas no ambiente desenvolvido.

No capítulo 5, apresentam-se as conclusões, evidenciando os avanços obtidos e identificando os possíveis desafios para o futuro.

2 O Protocolo DHCP

2.1 Descrição Sucinta

O DHCP permite especificar os parâmetros de rede de uma determinada sub-rede e foi especificado segundo o modelo cliente-servidor. Assim, o servidor DHCP especifica os parâmetros de rede dos clientes DHCP e fornece-lhes essa informação de forma automática a pedido.

Este protocolo tem como objectivo permitir a configuração de parâmetros de rede em máquinas remotas. É constituído por dois componentes (3):

a) Um protocolo que entrega os parâmetros de rede específicos para cada cliente a partir de um servidor DHCP (composto por vários serviços)

b) Um mecanismo dinâmico de atribuição de endereços de rede às diversas máquinas clientes.

Existem três mecanismos de atribuição de endereços (3):

a) Atribuição **Automática** – Aqui o protocolo atribui um endereço de rede a uma máquina cliente de forma permanente.

b) Atribuição **Dinâmica** – Aqui o protocolo atribui um endereço de rede a uma máquina cliente por um determinado período de tempo predefinido, salvo caso tal que, o cliente deliberadamente o solicite antes desse período terminar. É importante referir que a atribuição dinâmica também decorre de forma automática.

c) Atribuição **Manual** – Neste caso, o administrador do serviço DHCP atribui o endereço de rede manualmente, e o protocolo simplesmente envia essa informação para a máquina cliente remota.

A atribuição dinâmica é a única que permite a reutilização automática de endereços. Logo, torna-se particularmente útil para atribuir endereços de rede a máquinas que se ligam temporariamente a uma determinada sub-rede. É uma boa opção para redes cujo número de endereços IP disponíveis seja insuficiente para todas as suas potenciais máquinas, sendo assim necessário o reaproveitamento dos endereços daquelas que se desligam da rede, em favor das que se ligam.

Por todas estas razões, o sistema de atribuição dinâmica é o mais usado, logo será este o contexto do assunto em estudo.

2.2 Descrição Detalhada

O DHCP foi concebido para fornecer aos clientes DHCP diversos parâmetros de configuração da rede (5) tais como o endereço IP, máscara de sub-rede ou a porta de ligação predefinida, só para mencionar os mais importantes.
Após obter os parâmetros via DHCP, o cliente deve ser capaz de trocar informação com qualquer outra máquina da rede.

No entanto, nem todos os parâmetros são necessários para que o cliente DHCP inicialize e estabilize a ligação. A negociação entre cliente e servidor pode ocorrer para que sejam fornecidos apenas os parâmetros requisitados pelo cliente, ou referentes a uma determinada sub-rede.

2.2.1 Objectivos mais relevantes do protocolo DHCP

- O DHCP deve ser um mecanismo e não uma política. O serviço deve permitir aos administradores de rede controlo sobre os parâmetros de configuração.
- Os clientes do serviço não devem ter a necessidade de configuração manual. Os clientes devem descobrir por si próprios, os parâmetros de configuração adequados à rede em que se encontram.
- O serviço não requer um servidor em cada sub-rede. As mensagens DHCP propagam-se através dos encaminhadores (routers) recorrendo aos agentes de reencaminhamento (BOOTP[4] relay agents) permitindo assim a gestão centralizada.
- O cliente DHCP permite a recepção de múltiplas mensagens, permitindo assim a existência de vários servidores (e.g. por motivos de redundância).
- O serviço tem de garantir que não existam endereços de rede iguais em clientes DHCP diferentes.
- O serviço deve ser imune a reinicializações dos clientes DHCP, mantendo-lhes sempre que possível, os mesmos parâmetros de rede.
- O serviço também deve ser imune a reinicializações dos servidores, mantendo este o registo das configurações dos clientes.
- O serviço deve permitir a alocação automática a novos clientes DHCP, para evitar configuração manual.

[4] Bootstrap Protocol – O protocolo BOOTP está na origem do protocolo DHCP, sendo que este último garante o normal funcionamento de qualquer agente BOOTP.

2.2.2 Repositório dos parâmetros de configuração

O primeiro serviço que o DHCP implementa, é a capacidade de armazenar de forma persistente os parâmetros de configuração dos vários clientes. Para cada cliente tem de ser guardado um par chave-valor, a chave tem de ser única na sub-rede (e.g. endereço IP - endereço MAC) e o valor tem de conter os parâmetros de configuração respectivos.

2.2.3 Atribuição Dinâmica

O segundo serviço fornecido pelo DHCP é a atribuição temporária ou permanente de endereços de rede aos clientes (como referido antes, no nosso contexto desprezamos a atribuição automática e a manual).

Basicamente o mecanismo é o seguinte:

- O cliente requisita a atribuição de um endereço por um determinado intervalo de tempo – concessão (lease)
- O servidor fica obrigado a não atribuir esse mesmo endereço a outro cliente durante esse intervalo de tempo
- Cada vez que o cliente requisitar a extensão do tempo da concessão, o servidor atribuir-lhe-á o mesmo endereço de rede
- Quando o cliente não mais necessitar do endereço deve informar o servidor para proceder à libertação do endereço (termina a concessão)
- Se o cliente não efectuar este pedido, o servidor libertará o endereço a partir do momento em que o intervalo de tempo da concessão se esgota

  Para verificação de consistência da informação, o servidor deve verificar que o endereço que vai reatribuir, está disponível, assim como o cliente deve verificar que, de facto, o endereço que lhe vai ser atribuído, não está a ser usado (e.g. ARP[5]).

[5] Address Resolution Protocol

2.2.4 Mensagens DHCP

A Tabela 1 ilustra as mensagens DHCP existentes e respectiva descrição.

Tabela 1: Lista de Mensagens DHCP

DHCPDISCOVER	Mensagem de difusão do cliente para localizar servidores
DHCPOFFER	Mensagem do servidor para o cliente, como resposta a um DHCPDISCOVER, oferecendo parâmetros de configuração
DHCPREQUEST	Mensagem do cliente para os servidores para: pedir parâmetros oferecidos por um servidor, rejeitando as ofertas de outros ou confirmar a exactidão de parâmetros atribuídos anteriormente ou pedir a extensão da concessão (lease) para um determinado endereço de rede
DHCPACK	Mensagem do servidor para o cliente contendo os parâmetros de rede anteriormente oferecidos
DHCPNAK	Mensagem do servidor para o cliente, indicando que o cliente tem um endereço incorrecto ou que a sua concessão (lease) expirou
DHCPDECLINE	Mensagem do cliente para o servidor, indicando que o endereço de rede oferecido já está a ser usado
DHCPRELEASE	Mensagem do cliente para o servidor, indicando que já não pretende continuar a usar o endereço que lhe foi atribuído
DHCPINFORM	Mensagem do cliente para o servidor, pedindo informação sobre os parâmetros locais e informando que os seus parâmetros já foram configurados externamente

2.2.5 Processo de atribuição de um endereço de rede

A Figura 1 ilustra o diagrama temporal do processo que se passa a descrever:

1 O cliente emite em difusão (broadcast) para a rede uma mensagem DHCPDISCOVER. Esta mensagem pode incluir opções que sugerem valores para o endereço IP e para o tempo de duração da concessão (lease). Caso o servidor faça a gestão de outras redes, os agentes de reencaminhamento (relay agents) BOOTP/DHCP são responsáveis por fazer chegar as mensagens às mesmas (isto é válido para todas as mensagens DHCP).

2 Cada servidor pode responder com uma mensagem DHCPOFFER que inclui um endereço de rede disponível, além de outros parâmetros de configuração. O servidor não tem a obrigação de reservar de imediato o endereço, mas pode fazê-lo e recomenda-se que o faça, estando provado que desta forma é mais eficiente (3). Quando efectivamente, o servidor proceder à atribuição do endereço a um cliente, deve verificar que o endereço não foi já atribuído (e.g. pedido de eco (echo request) ICMP).

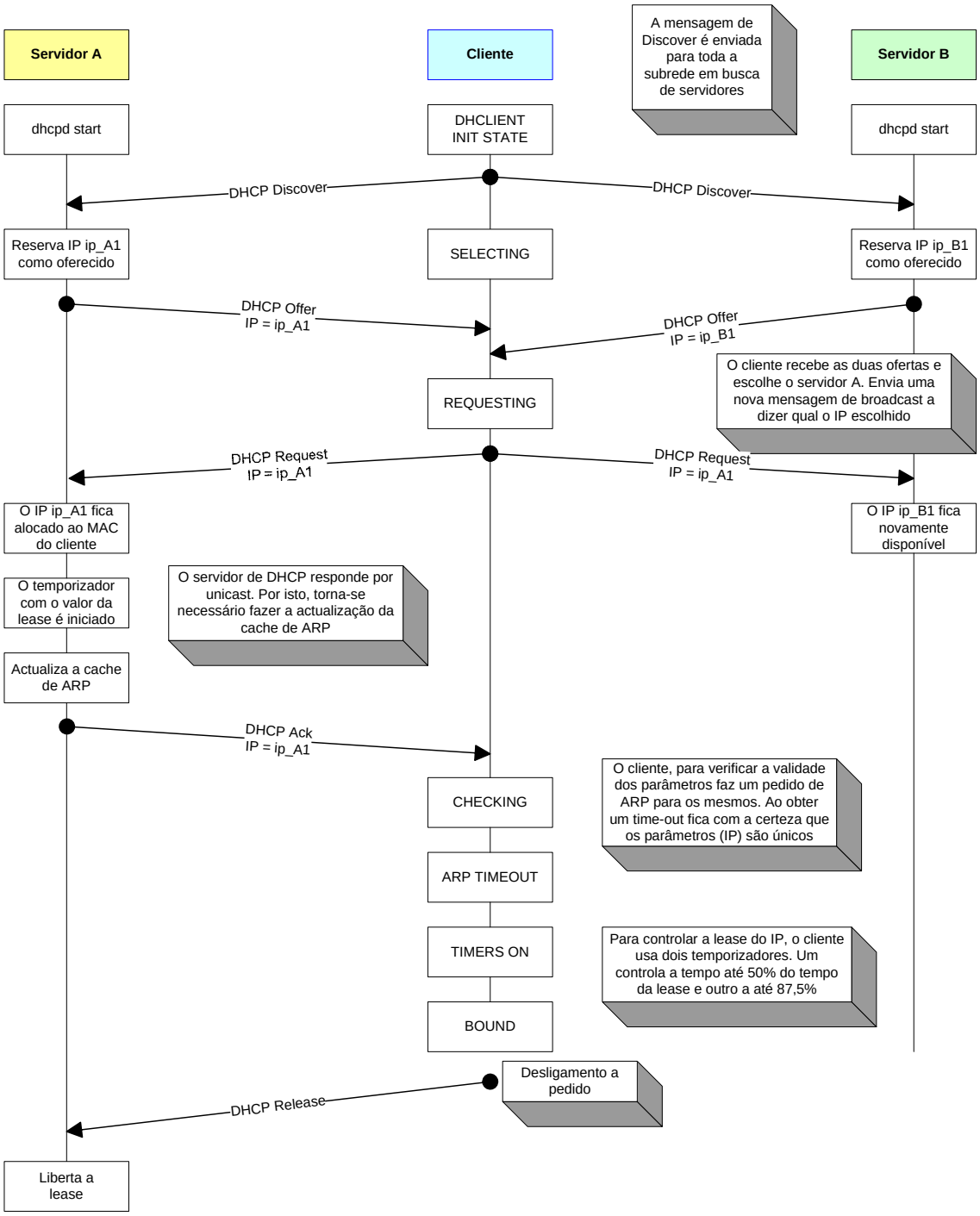

Figura 1: Diagrama temporal na atribuição de um endereço de rede

3 O cliente recebe uma ou mais mensagens DHCPOFFER de um ou mais servidores. O cliente pode escolher se espera ou não por múltiplas respostas. O cliente escolhe então, qual o servidor ao qual vai requisitar os parâmetros de configuração, baseando-se nos parâmetros de configuração existentes nas mensagens DHCPOFFER. Após seleccionado, o cliente emite uma mensagem de difusão DHCPREQUEST que tem de incluir a opção "server identifier" (ver Tabela 3: Lista de opções relevantes) com o servidor previamente seleccionado e pode ainda, incluir outras opções especificando valores desejados de configuração.

4 Os servidores recebem a mensagem DHCPREQUEST do cliente.

Os servidores não seleccionados usam a mensagem para perceber que o cliente rejeitou a sua oferta.
O servidor seleccionado armazena a informação de forma definitiva (armazenamento persistente) e envia uma mensagem DHCPACK ao cliente, onde inclui os parâmetros de configuração requisitados. O par constituído pelo endereço de rede e o identificador do cliente tem de ser único e fica associado à concessão (lease) do cliente para futuras mensagens DHCP. Os parâmetros incluídos na mensagem DHCPACK, não devem de forma alguma, ser diferentes dos parâmetros previamente incluídos na mensagem DHCPOFFER. Se por acaso, o servidor não conseguir atender ao pedido do cliente, deve responder ao mesmo com uma mensagem DHCPNAK.

5 O cliente recebe a mensagem DHCPACK com os parâmetros de configuração. O cliente deve verificar os parâmetros recebidos através de uma mensagem de ARP (para o endereço de rede) e se o tempo de concessão (lease) é o mesmo que constava na mensagem DHCPOFFER. Neste momento o cliente fica configurado (estado BOUND). Se o cliente detectar que o endereço já está a ser utilizado (e.g. através de ARP), deve informar o servidor através de uma mensagem DHCPDECLINE e recomeçar o processo de configuração. O cliente deve aguardar um mínimo de 10 segundos antes de recomeçar o processo de configuração para evitar tráfego excessivo na rede devido a ciclos infinitos (looping). Se o cliente receber uma mensagem DHCPNAK recomeça o processo de configuração. O cliente espera que um determinado tempo se esgote 3(timeout) e retransmite a mensagem DHCPREQUEST se não receber qualquer uma das mensagens DHCPACK ou DHCPNAK. Após várias retransmissões (ver página 29) o cliente desiste e recomeça o processo de configuração. Neste momento, o cliente deve notificar o utilizador que o processo de configuração falhou.

6 O cliente pode decidir libertar o seu endereço enviando ao servidor uma mensagem DHCPRELEASE. O cliente identifica a concessão (lease) a ser libertada com o par "client identifier" e "hardware address" (ver secção 2.2.7).

2.2.6 Processo de reutilização de um endereço de rede

Se o cliente pretender reutilizar um endereço previamente utilizado, pode fazê-lo omitindo alguns dos passos referidos na secção anterior (ver Figura 2).

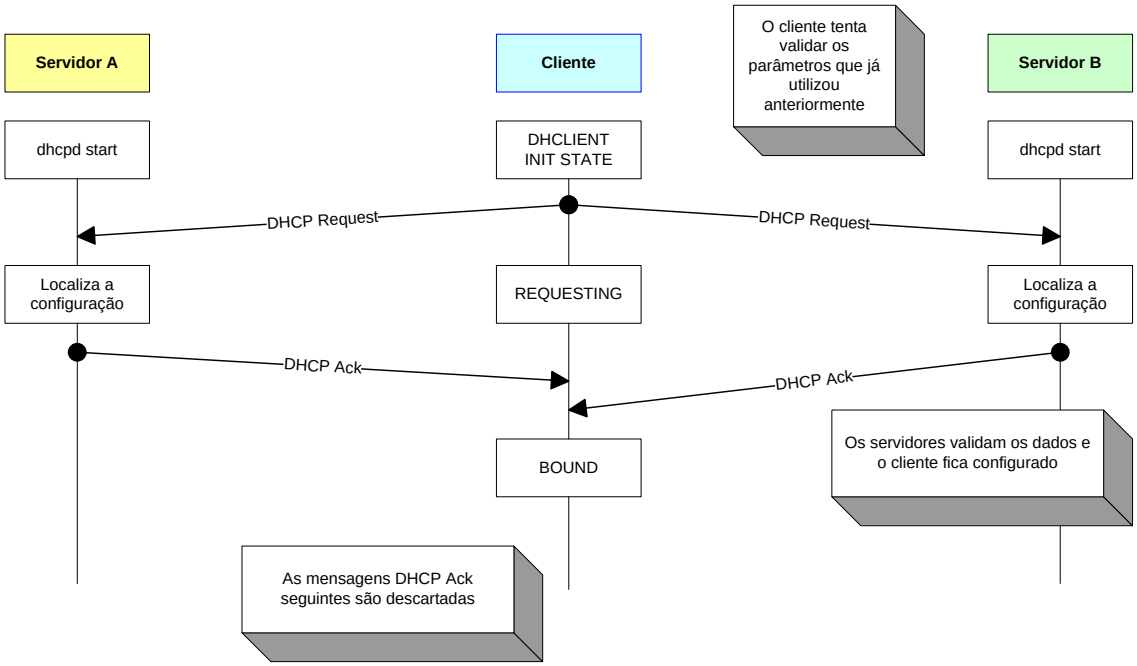

Figura 2: Diagrama temporal na reutilização de um endereço

1 O cliente emite em difusão (broadcast) uma mensagem DHCPREQUEST dentro da sua sub-rede. A mensagem inclui o endereço de rede do cliente e o endereço dos agentes de reencaminhamento (relay agents) BOOTP, se existirem, para que estes possam fazer chegar esta mensagem a outros servidores DHCP noutras sub-redes.

2 Os servidores que possuírem os parâmetros de configuração do cliente em causa respondem com uma mensagem DHCPACK. Aqui os servidores não devem verificar se o endereço já está a ser utilizado, pois o cliente que fez o pedido também responderia ao pedido de verificação (e.g. ICMP).
Se o pedido do cliente é inválido (e.g., está noutra sub-rede fora das gamas de endereçamento), os servidores respondem com uma mensagem DHCPNAK para o cliente.

3 O cliente recebe a mensagem DHCPACK com os parâmetros de configuração. O cliente procede a uma verificação e actualiza o valor da concessão (lease). Neste momento, o cliente está configurado (estado BOUND).

   Se o cliente detectar que o endereço na mensagem já está a ser utilizado, envia uma mensagem DHCPDECLINE para o servidor e recomeça o processo de configuração.

   Caso o cliente receba uma mensagem DHCPNAK, não pode reutilizar o endereço que em tempos possuiu. O cliente, neste momento, recomeça o processo de configuração normal descrito na secção anterior (ver página 16).

   Por último, se o cliente não receber qualquer uma das mensagens já referidas, retransmite o pedido um predeterminado número de vezes. Após isso, desiste e pode optar por manter o endereço que tinha anteriormente, pois a concessão (lease) não expirou.

4 Aqui, também o cliente pode optar por libertar a sua concessão (lease) enviando uma mensagem DHCPRELEASE. Fá-lo deliberadamente se for necessário mudar a máquina para outra sub-rede ou se a desligar.

2.2.7 Construção e envio de mensagens DHCP

Os clientes e os servidores DHCP constroem mensagens preenchendo campos (ver Tabela 2) num formato específico (Figura 3).

Tabela 2: Campos de uma mensagem DHCP

Campo	Octetos	Descrição
op	1	Tipo de mensagem 1 - BOOTREQUEST 2 - BOOTREPLY
htype	1	Tipo de endereço de hardware
hlen	1	Comprimento do endereço de hardware
hops	1	O cliente coloca a zero. Os "relay agents" incrementam este campo
xid	4	Identificador da transacção
secs	2	É preenchido pelo cliente. Conta os segundos desde que o cliente inicia um processo de aquisição ou renovação de endereços
flags	2	B|MBZ
ciaddr	4	Endereço IP do cliente
yiaddr	4	Endereço IP do cliente (your IP address)
siaddr	4	Endereço IP do próximo servidor a usar no processo de arranque
giaddr	4	Endereço IP do "relay agent" se tal foi usado no processo de inicialização
chaddr	16	Endereço de hardware do cliente
sname	64	Nome do servidor (opcional)
file	128	Ficheiro de inicialização
options	>312	Parâmetros opcionais

Um dos campos é variável (options). Este campo permite incluir opções que já estão definidas (6) assim como incluir opções novas no futuro. A lista de opções é imensa, mas fica aqui uma lista que inclui as que são relevantes para este trabalho, pois serão incluídas na construção de mensagens DHCP, na solução proposta (consultar (6) para ver lista completa).

Tabela 3: Lista de opções relevantes

Requested IP address	Usada pelos clientes nas mensagens DHCPDISCOVER quando querem especificar um endereço
IP address (lease) time	Especifica o tempo da concessão pedido ou atribuído
DHCP message type	Define o tipo de mensagem (e.g. 1 - DHCPDISCOVER)
Parameter request list	Usada pelos clientes para definir parâmetros específicos pretendidos
Message	Serve complementar a informação (e.g. justificar os erros)
Client identifier	Usado pelo cliente para se identificar. Os servidores mapeiam os endereços por este campo (e.g. htype/chaddr)
Server identifier	Usada nas mensagens DHCPOFFER e DHCPREQUEST, e opcionalmente nas mensagens DHCPACK ou DHCPNAK. Os servidores incluem esta opção para permitir ao cliente distinguir entre ofertas de concessão. Os clientes usam para enviar mensagens *unicast* ao servidor. Também serve aos clientes para, nas mensagens DHCPREQUEST, definirem qual o servidor cuja oferta aceitaram.
Maximum message size	Usada pelos clientes para limitar o número de opções que os servidores pretendam incluir
Renewal (T1) Time Value	Valor do temporizador 1 (50%)
Rebinding (T2) Time Value	Valor do temporizador 2 (87,5%)

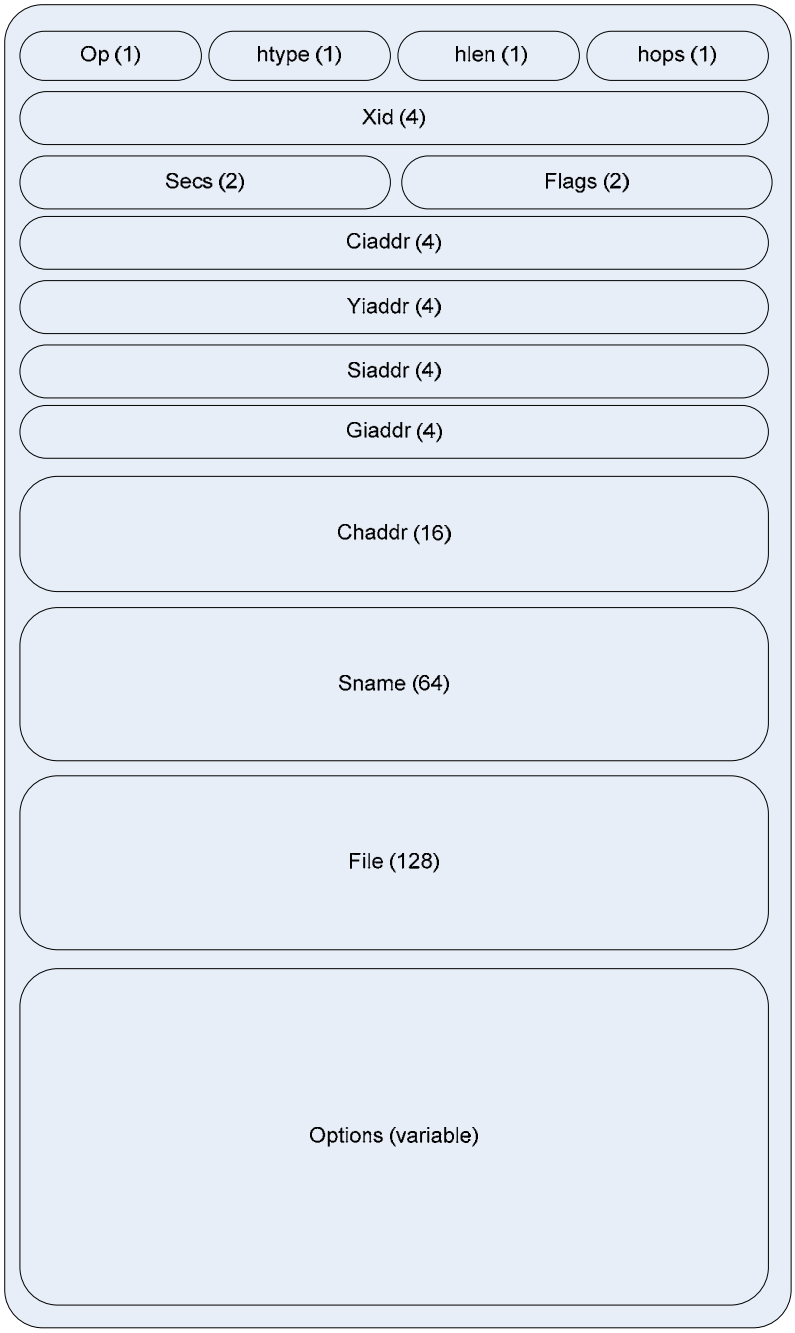

Figura 3: Formato de uma mensagem DHCP

2.2.7.1 Protocolo de transporte

O DHCP usa UDP como protocolo na camada de transporte. As mensagens enviadas do cliente para o servidor são enviadas para o porto 67 do servidor e as mensagens do servidor para o cliente são enviadas para o porto 68 do cliente.

Os clientes são responsáveis por todas as retransmissões de mensagens. O cliente adopta uma estratégia de retransmissão que inclua um algoritmo para determinar o atraso entre retransmissões. Esse atraso é escolhido de forma a

dar tempo ao servidor de responder, baseando-se nas características da rede que interliga o cliente e o servidor.

2.2.7.2 Mensagens de difusão (broadcast) e de destino definido (unicast)

É importante referir como é que o protocolo DHCP distingue entre mensagens de difusão e de não–difusão.

A mensagem é de difusão quando é enviada para todas as máquinas da sub-rede em que a máquina que envia a mensagem se encontra (chega a outras sub-redes (se for caso disso) através dos agentes de reenvio (relay agents) presentes nos encaminhadores (routers)).

Para que tal aconteça, o DHCP usa o campo “flags” (ver Figura 3). O primeiro bit do primeiro octeto (ver Figura 4), é usado para definir se a mensagem é de difusão ou não (1 ou 0). Os restantes bits são colocados a zero pelos clientes e são ignorados por servidores e agentes, estando reservado para uso futuro.

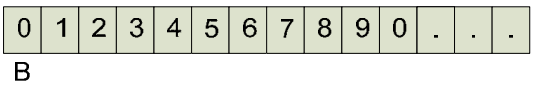

Figura 4: "Flag" de difusão (broadcast)

2.2.8 Comportamento do servidor DHCP

O servidor DHCP processa as mensagens DHCP enviadas pelos clientes à medida que forem chegando. O servidor pode receber as seguintes mensagens:

a) DHCPDISCOVER
b) DHCPREQUEST
c) DHCPDECLINE
d) DHCPRELEASE
e) DHCPINFORM

Estas mensagens são construídas pelos clientes DHCP, preenchendo os campos da Tabela 2. Estas mensagens têm o formato da Figura 3.

a. Mensagem DHCPDISCOVER

Quando um servidor recebe uma mensagem DHCPDISCOVER de um cliente, escolhe um endereço de rede para o cliente em causa.

Se não tiver qualquer endereço disponível, o servidor avisa o administrador do sistema que tal ocorreu.

Se tiver endereços disponíveis, o servidor deve escolher o endereço a atribuir segundo os seguintes critérios por ordem de prioridade:

- O endereço actualmente registado para este cliente
- O endereço previamente registado para este cliente, se ainda estiver disponível e pertencer ao conjunto de endereços que este servidor tem para disponibilizar
- O endereço que conste na opção "Requested IP address" da mensagem DHCPDISCOVER, sendo este válido e estiver disponível
- Um endereço novo, escolhido do conjunto de endereços fornecidos pelo servidor. É escolhido um pertencente à sub-rede do remetente da mensagem (se o campo "giaddr" for 0) ou pertencente à sub-rede do agente de reencaminhamento (relay agent) que remete a mensagem (caso o campo "giaddr" seja 1)
- O servidor pode ainda escolher outro endereço qualquer por questões de administração do serviço, ou até recusar a atribuição do endereço. A forma como estas situações são geridas, é independente do protocolo DHCP e depende da configuração do serviço

O servidor também tem de escolher um tempo de expiração da concessão (lease), usando os seguintes critérios:

- Se o cliente não especificou uma concessão (lease) na mensagem DHCPDISCOVER (isto implica que não é um pedido de extensão) e já teve um endereço de rede atribuído, então o servidor atribui-lhe o tempo de expiração igual ao que tinha sido atribuído anteriormente.
- Se o cliente não especificou uma concessão (lease) e ainda não teve qualquer endereço de rede anteriormente atribuído, o servidor atribui o tempo de expiração configurado localmente
- Se o cliente especificar uma concessão (lease) na mensagem DHCPDISCOVER, independentemente de já ter tido ou não um endereço de rede atribuído, o servidor pode aceitar ou atribuir outra. A escolha depende da política local

Determinados o endereço de rede e a concessão (lease), o servidor constrói uma mensagem DHCPOFFER incluindo os parâmetros designados.

É importante que todos os servidores DHCP retornem os mesmos parâmetros, mediante as mesmas condições, para que os clientes DHCP tenham um comportamento determinístico independentemente do servidor que seleccionarem.

Os parâmetros de configuração devem ser seleccionados seguindo os critérios abaixo apresentados, pela ordem de apresentação. O servidor tem de retornar:

- O endereço de rede do cliente, conforme descrito anteriormente nesta secção

- O tempo de expiração da concessão (lease) do cliente, conforme descrito anteriormente nesta secção
- Os parâmetros requisitados pelo cliente, de acordo com as seguintes regras:
    - Se o servidor foi explicitamente configurado com um valor por defeito para esse parâmetro, então tem de incluir esse valor na opção "server identifier"
    - Se o servidor reconhecer o parâmetro como pertencente ao documento "Host Requirements Document", o servidor tem de incluir o valor por defeito desse parâmetro especificado no documento na opção apropriada do campo "options"
    - Caso não se verifique nenhuma das situações anteriores, o servidor não deve retornar qualquer valor para o parâmetro

  O servidor deve retornar o máximo de parâmetros que conseguir atribuir e deve omitir aqueles que não consegue atribuir.

- Quaisquer parâmetros da atribuição existente, que sejam diferentes dos valores por defeito
- Quaisquer parâmetros específicos deste cliente (conforme o conteúdo de "chaddr" ou "client identifier") configurados pelo administrador de rede
- Quaisquer parâmetros especificados nas "options" ("vendor class identifier")
- Os parâmetros com valores diferentes dos valores por defeito para uma determinada sub-rede

b. Mensagem DHCPREQUEST

Uma mensagem DHCPREQUEST pode ter como remetente um cliente a responder a uma mensagem DHCPOFFER, um cliente a verificar um endereço de rede previamente atribuído ou um cliente a pedir a extensão da concessão (lease) de um endereço.

Se a mensagem incluir a opção "server identifier", então esta mensagem representa uma resposta a uma mensagem DHCPOFFER.

Se não incluir, então é uma verificação ou um pedido de extensão de uma concessão (lease) existente.

Se o cliente usar a opção "server identifier" na mensagem DHCPREQUEST, tem de usar a mesma opção em todas as mensagens seguintes.

Se o cliente requisitar uma lista de parâmetros na mensagem DHCPDISCOVER, tem o fazer em todas as mensagens seguintes.

O servidor tem de ser coerente. Todos os parâmetros presentes numa mensagem DHCPACK, não podem de forma alguma ser diferentes dos parâmetros presentes na mensagem DHCPOFFER à qual o cliente está a

responder. O cliente tem de usar os parâmetros presentes no DHCPOFFER para a configuração.

Dependendo do estado em que se encontra, o cliente constrói a mensagem DHCPREQUEST da seguinte forma:

- Estado SELECTING

  O cliente preenche o a opção "server identifier" com o endereço do servidor seleccionado. O campo "ciaddr" tem de ser 0. A opção "endereço IP requisitado" é preenchido com o valor que constar no campo "yiaddr" da mensagem DHCPOFFER seleccionada.

  Caso nenhuma oferta seja seleccionada, os servidores não irão receber nenhuma mensagem DHCPREQUEST a informá-los disso. Os servidores devem possuir um mecanismo de controlo temporal (timeout) específico da implementação para lidar com esta situação.

- Estado INIT-REBOOT

  A opção "server identifier" deve ser deixado vazia e a opção "Requested IP Address" deve ser preenchida com o valor que o cliente possuir (ou entender possuir) como o seu anterior endereço de rede. O campo "ciaddr" deve ser 0.

  Nesta situação o cliente procura verificar os parâmetros que possui de uma anterior configuração.

  Se o servidor determinar que tais parâmetros são incorrectos (e.g. pertencentes a outra sub-rede), deve enviar uma mensagem DHCPNAK ao cliente para que este reinicie o processo de configuração.

  Se o servidor não reconhecer este cliente (não possuir nenhum registo anterior) não deve enviar qualquer resposta. Isto é muito importante para que vários servidores DHCP que não comuniquem entre si, coexistam pacificamente.

  Se o servidor determinar que os parâmetros estão correctos, envia uma mensagem DHCPACK para o cliente.

- Estado RENEWING

  Nesta situação a opção "server identifier" deve ficar vazia assim como a opção "Requested IP Address", enquanto que o campo "ciaddr" deve ser preenchido com o endereço IP do cliente.

  O cliente está configurado e pretende uma extensão da concessão (lease). Se o pedido for anterior a T1 cabe ao administrador de rede definir na configuração se pretende que tais pedidos sejam atendidos. De qualquer forma o servidor tem de enviar uma mensagem DHCPACK para o cliente.

- Estado REBINDING

  A opção “server identifier” não pode ser preenchida, assim como a opção “Requested IP Address”. O “ciaddr” é preenchido com o endereço IP do cliente.

  Aqui o cliente, pretende mais uma vez, a extensão da sua concessão (lease). O servidor deve verificar a veracidade do endereço do cliente antes de responder.

c. Mensagem DHCPDECLINE

Se o servidor receber uma mensagem DHCPDECLINE isso significa que o cliente descobriu por outros meios que o endereço de rede sugerido (na mensagem DHCPOFFER anterior) já está a ser utilizado. Assim, o servidor tem de marcar este endereço como não disponível e deve avisar o administrador de rede para um possível problema de configuração.

d. Mensagem DHCPRELEASE

Ao receber uma mensagem DHCPRELEASE o servidor marca o endereço de rede que consta na mensagem como não atribuído. O servidor deve guardar os parâmetros de configuração do cliente, no caso de haver um pedido do cliente por estes parâmetros *a posteriori*.

e. Mensagem DHCPINFORM

O servidor responde a uma mensagem DHCPINFORM com uma mensagem DHCPACK para o endereço que consta no campo “ciaddr” da mensagem DHCPINFORM. Nesta mensagem não pode constar o tempo de expiração da concessão (lease) assim como tem de estar vazio o campo “yiaddr”.

2.2.9 Comportamento do cliente DHCP

O cliente pode receber as seguintes mensagens do servidor

- DHCPOFFER
- DHCPACK
- DHCPNAK

Ao receber cada uma destas mensagens, o cliente comporta-se segundo o diagrama de estados da Figura 5.

2.2.9.1 Inicialização e atribuição de endereços de rede

O cliente ao arrancar constrói imediatamente uma mensagem DHCPDISCOVER. Esta mensagem pode ou não incluir, parâmetros pedidos especificamente pelo utilizador. Esta mensagem é emitida em difusão pela sub-rede.

O cliente fica a aguardar e colecciona as mensagens DHCPOFFER enviadas pelos servidores durante um período de tempo após o qual, selecciona uma segundo um determinado critério. Como este período é calculado, depende da implementação.

Sendo os parâmetros aceitáveis, o cliente extrai a informação sobre o servidor da mensagem DHCPOFFER e constrói uma mensagem DHCPREQUEST. Esta mensagem, é também ela, difundida pela sub-rede, para informar todos os servidores, qual o servidor escolhido. Quando receber a mensagem DHCPACK do servidor a confirmar o pedido o cliente fica então inicializado.

O cliente guarda o tempo de expiração da concessão (lease), não deixando de verificar a validade dos parâmetros recebidos. Se os parâmetros forem inválidos por alguma razão, o cliente envia uma mensagem DHCPDECLINE ao servidor.

2.2.9.2 Inicialização com endereço de rede já conhecido

O cliente constrói a mensagem DHCPDISCOVER incluindo já o endereço de rede pretendido (provavelmente, anteriormente atribuído). A única diferença relativa ao ponto anterior, é que o cliente provavelmente vai usar os parâmetros que já conhece no processo de selecção das mensagens DHCPOFFER que recolheu.

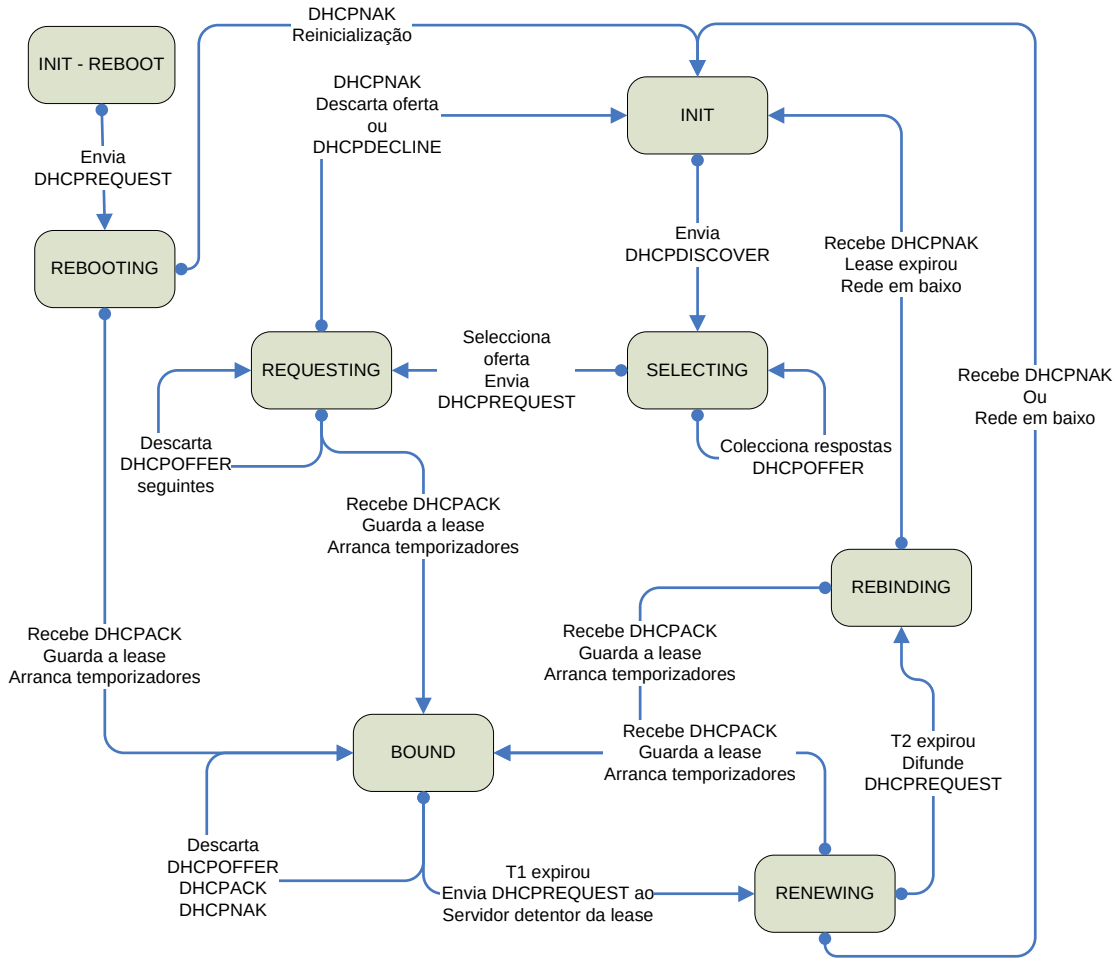

Figura 5: Diagrama de estados do cliente DHCP

2.2.9.3 Inicialização com configuração manual

O cliente envia uma mensagem DHCPINFORM com os parâmetros que já possui e não pede quaisquer parâmetros aos servidores. O DHCPACK que deve receber serve para confirmar que o servidor não encontrou conflitos entre a sua configuração e a configuração do cliente. Após a recepção desta mensagem o cliente fica inicializado. Se não receber a mensagem após um determinado período de tempo, o cliente informa o utilizador que não é possível usar aquela configuração.

2.2.9.4 Gestão temporal

O cliente mantém dois temporizadores, T1 e T2, que especificam quando é que o cliente vai tentar estender o tempo da sua concessão (lease). T1 representa o instante em que o cliente tenta renovar os parâmetros que já possui e T2 representa o instante em que o cliente tenta renovar os parâmetros, mesmo que não sejam os que já possui (e.g., recorrendo a outro servidor DHCP). Para evitar a necessidade de sincronização, T1 e T2 são

tempos relativos. Estes tempos são configuráveis, correspondendo por defeito a 50% e a 87,5% do tempo da concessão (lease), para T1 e T2 respectivamente.

2.2.9.5 DHCPRELEASE

Se o cliente não mais necessitar do endereço que lhe foi atribuído (e.g., vai desligar-se por opção do utilizador), envia uma mensagem DHCPRELEASE ao servidor. O envio ou não desta mensagem não afecta o correcto funcionamento do protocolo.

2.2.10 Campo "options" das mensagens DHCP

É de todo o interesse referir aqui algumas das opções das mensagens, que serão relevantes para o nosso trabalho.

Para além dos parâmetros que, por questões protocolares têm de obrigatoriamente constar nas mensagens, podemos usar os seguintes campos para atingir o fim que pretendemos (cap. 9, (6)), que é conhecer o estado do serviço por observação do tráfego:

1 Server Identifier

Pode fazer parte de qualquer uma das mensagens enviadas pelo servidor e é usada pelos clientes DHCP para saber a que servidor devem mandar mensagens unicast. É preenchido com o endereço IP do servidor DHCP.

2 Renewal Time Value (T1)

Como é um valor configurável do lado do servidor, é importante capturá-lo, para fins de cálculo de tempos esperados

3 Rebinding Time Value (T2)

Como é um valor configurável do lado do servidor, é importante capturá-lo, para fins de cálculo de tempos esperados

4 Client Identifier

Opção usada como identificador único dos clientes DHCP. Pode ser útil para fins de classificação dos clientes DHCP

É possível definir novas extensões ao campo das opções do DHCP. Poderá ser uma alternativa no caso de se verificar que alguma parte do algoritmo não possa ser implementada por falta de dados a circular nas mensagens.

3 Cenários de Utilização

Este capítulo tem como objectivo descrever os vários cenários reais de utilização do serviço DHCP e as várias soluções de redundância já utilizadas para aumentar a fiabilidade do serviço.

Dividiu-se esta análise em dois grandes grupos. Primeiro serão considerados os cenários de simples utilização do serviço na ausência de falhas. Posteriormente contemplar-se-á a existência de falhas fazendo uma análise a várias possíveis soluções de redundância.

Em cada cenário será efectuada uma análise de fiabilidade e no fim teremos um quadro comparativo entre os vários cenários.

Uma análise de fiabilidade pressupõe a definição de um intervalo de tempo até a ocorrência de uma falha (7). No entanto, como para este fim o que pretendemos é comparar os diversos cenários, considera-se o intervalo de tempo como indefinido mas igual para todos os cenários. Assim para efeitos comparativos podemos desprezar o factor tempo.

Um sistema tão complexo como uma rede de computadores/máquinas possui inúmeros pontos de falha. Por questões de simplificação da análise iremos reduzir para duas as possibilidades de falha. Consideramos que a máquina cliente não falha, pois se tal acontecesse, não teria necessidade do serviço. Assim podemos concentrar os pontos de falha em:

- Falha de servidor (por motivos de hardware ou software)
- Falha de conectividade (motivos de hardware ou software)

O primeiro (falha de servidor) traduz o conjunto de possibilidades de falha que poderão ter origem no extremo do sistema (e.g. máquina em baixo por avaria, serviço em baixo, serviço mal configurado, etc.).
O segundo (falha de conectividade) traduz o conjunto de possibilidades de falha que possam ocorrer no percurso entre os extremos do sistema. A Figura 6 traduz esta simplificação.

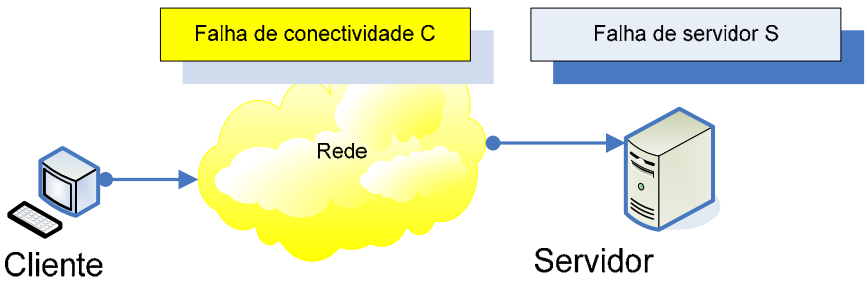

Figura 6: Análise de Fiabilidade Simplificada

3.1 Cenários sem redundância

O serviço DHCP é utilizado pelos administradores de rede em vários contextos. Pode ser usado para a atribuição de endereços de rede numa rede local (LAN) ou numa rede alargada (WAN), sendo que os cenários de rede local podem ser logicamente distribuídos ou não.

A Tabela 4 mostra os vários casos que surgem, considerando que as redes se distribuem de uma forma lógica e geográfica.

Tabela 4: Cenários de Utilização

	Logicamente distribuída	**Geograficamente distribuída**
LAN	Não	Não
	Sim	Não
WAN	Sim	Sim

3.1.1 Análise de fiabilidade

Sem redundância, uma rede pode ser vista como um conjunto de equipamentos em série. A fiabilidade de uma rede deste tipo é representada pela fiabilidade do conjunto de equipamentos em série (a máquina cliente, o cabo que liga ao comutador, o comutador, o encaminhador (router), a WAN (se existir), o servidor, etc.).
Assim, basta que um equipamento da rede falhe, para que o serviço falhe (sistema série).

Considerando um sistema série simplificado de dois elementos, probabilidade do sistema falhar é igual à probabilidade de um dos elementos ou do outro falhar.

Sabemos que:

$$\text{(a)} \qquad P_f(a \cup b) = P_f(a) + P_f(b) - P_f(a \cap b)$$

e que,

$$\text{(b)} \qquad P_f(a \cap b) = P_f(a) \times P_f(b)$$

Porque os acontecimentos (falha de um dos elementos) são independentes.

Substituindo (b) em (a) temos,

$$\text{(c)} \qquad P_f(a \cup b) = P_f(a) + P_f(b) - P_f(a) \times P_f(b)$$

A fiabilidade de um elemento ou sistema é igual à probabilidade deste não falhar,

$$R = 1 - P_f \qquad \text{(d)}$$

Logo a probabilidade de falha do sistema ou elemento vem:

$$P_f = 1 - R \qquad \text{(e)}$$

Substituindo (d) e (e) em (c) obtém-se o seguinte desenvolvimento:

$$R = 1 - \left[(1 - R_a) + (1 - R_b) - \left[(1 - R_a) \times (1 - R_b)\right]\right]$$

$$R = 1 - \left[2 - R_a - R_b - 1 + R_a + R_b - R_a \times R_b\right]$$

$$R = 1 - \left(1 - R_a \times R_b\right)$$

Donde se conclui que a fiabilidade do sistema em série é igual ao produto das fiabilidades individuais de cada um dos elementos.

$$R = R_a \times R_b$$

Generalizando para x elementos fica:

$$R = \prod_{n=1}^{x} R_n \qquad \text{(f)}$$

Esta expressão será utilizada como base de discussão nas secções seguintes onde se consideram os vários cenários sem redundância.

3.1.2 Redes Locais (LAN)

Num contexto de rede local, o servidor DHCP reside numa máquina presente na mesma zona geográfica mas não necessariamente no mesmo segmento de rede.
De seguida estudamos o caso em que o cliente e o servidor se encontram dentro do mesmo segmento de rede, para de seguida estudarmos o caso em estes se encontram em segmentos de rede diferentes.

3.1.2.1 Clientes e servidores no mesmo segmento de rede

O caso mais simples é o da Figura 7, onde temos um servidor DHCP local, quer do ponto de vista geográfico, quer do ponto de vista lógico.

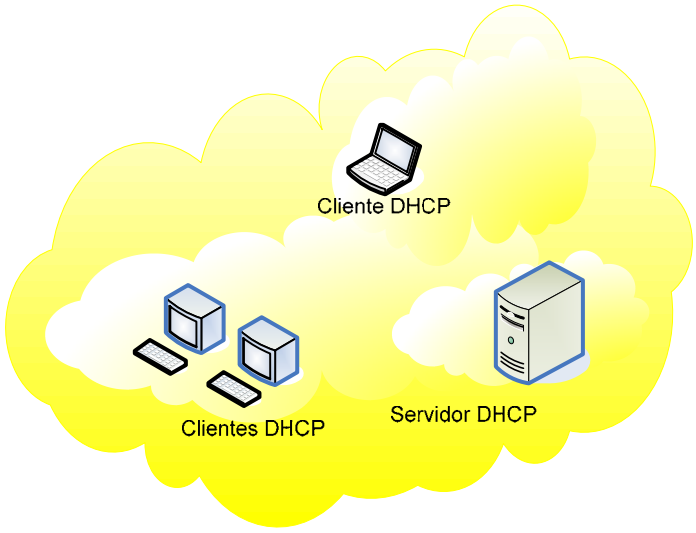

Figura 7: Rede local com um segmento de rede

Em termos de fiabilidade neste cenário podemos considerar que se trata de um sistema em série entre a conectividade e o servidor, pois basta que um falhe para que os clientes DHCP fiquem sem acesso ao serviço. Assim de (f) concluímos que a fiabilidade *R* vem:

$$R = (1-S)\times(1-C)$$

em que S é a probabilidade de falha do servidor e C a probabilidade de falha da conectividade.

A probabilidade de falha de conectividade C pode ainda ser dividida entre a probabilidade de falha dos equipamentos passivos p (e.g. cabos, fichas, etc.) e a probabilidade de falha dos equipamentos activos a (e.g. comutadores, encaminhadores, etc.).

Por uma questão de simplificação da análise assume-se que a probabilidade de falha de todos os equipamentos é igual, embora se saiba que na realidade tal não acontece.

Sendo n o número de equipamentos activos e m o número de equipamentos passivos entre os clientes e o servidor, a fiabilidade fica:

$$R = (1-S) \times \prod_{i=1}^{n} (1-a_i) \times \prod_{i=1}^{m} (1-p_i)$$

3.1.2.2 Clientes e servidor em segmentos de rede diferentes

Na Figura 8 ilustra-se a utilização do serviço DHCP numa rede local, onde os clientes não pertencem ao segmento de rede do servidor, i.e. pertencem a sub-redes diferentes (e.g. *campus*).

NOTA: O encaminhador (Router) deve conter um agente de encaminhamento de pacotes BOOTP (BOOTP Relay Agent).

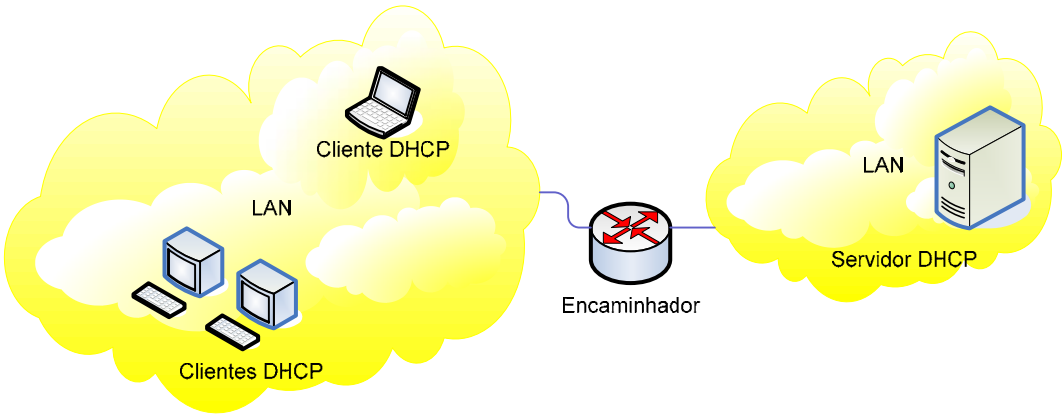

Figura 8: Rede local com o servidor num segmento de rede diferente

Em termos de fiabilidade, este cenário difere do anterior com a adição de mais pontos de falha no sub-capítulo dos equipamentos contribuindo para um aumento da probabilidade de falha C (a+p). Para um número adicional M de equipamentos passivos com uma probabilidade de falha p e um número adicional *N* de equipamentos activos com uma probabilidade de falha a, a fiabilidade *R* vem então:

$$R = (1-S) \times \prod_{i=1}^{n+N} (1-a_i) \times \prod_{i=1}^{m+M} (1-p_i)$$

De forma empírica sabe-se que a probabilidade de falha dos equipamentos passivos é relativamente baixa numa rede local e que, os equipamentos activos de alta fiabilidade decrescem continuamente de custo.
Assim, assumindo que numa rede não distribuída geograficamente o responsável tem um tempo de resposta curto para qualquer problema de configuração que surja, é espectável uma fiabilidade elevada do serviço DHCP numa rede local, tenha esta ou não mais do que um segmento de rede.

3.1.3 Redes Alargadas (WAN)

No contexto das redes alargadas, na generalidade das situações, as entidades interessadas (e.g. empresas ou universidades) optam por contratar os serviços de interligação entre as diversas sub-redes geograficamente distribuídas, a um operador de telecomunicações. De uma forma simples podemos afirmar que estas entidades recorrem à Internet para interligar as suas redes locais sendo este o cenário de utilização do serviço DHCP mais comum.

A excepção à regra encontra-se precisamente na utilização do serviço DHCP pelo próprio operador de telecomunicações (ISP). Ambos os casos são estudados nesta secção.

3.1.3.1 Rede Alargada com recurso à Internet

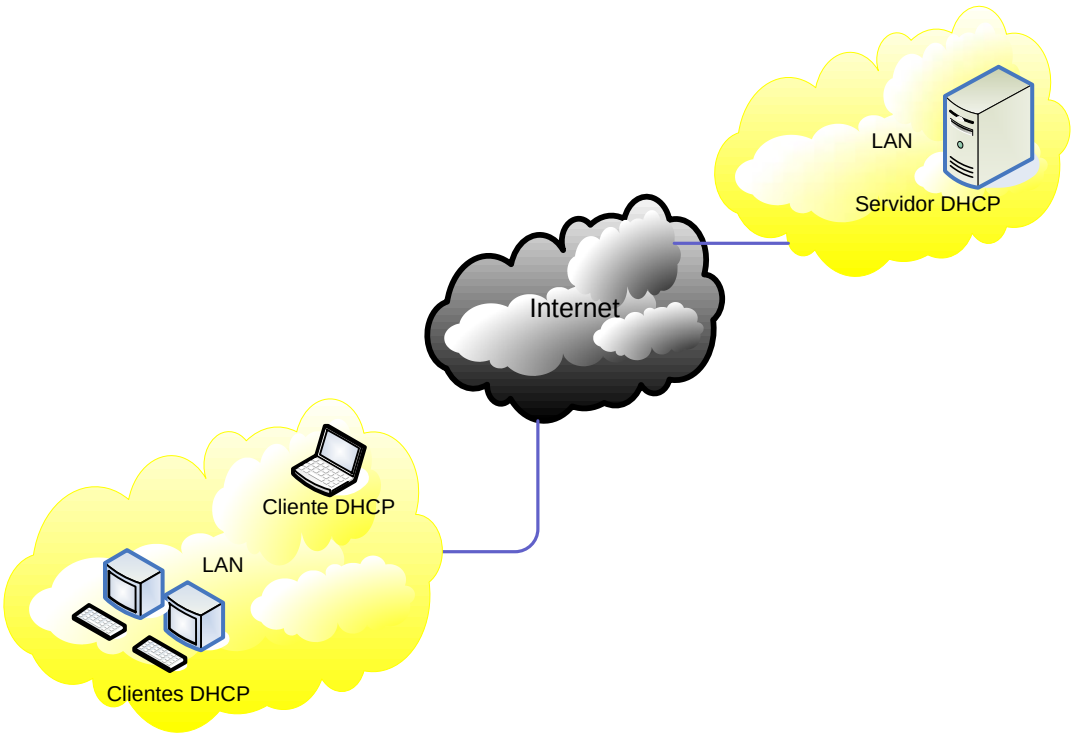

Figura 9: Rede Alargada com recurso à Internet

A Figura 9 mostra a utilização do serviço DHCP numa rede alargada que recorre à rede pública (Internet) para interligar as diferentes redes locais.

Em termos de fiabilidade este cenário muda completamente as probabilidades de falha. Ao invés de colocarmos um equipamento controlado por nós entre as sub-redes (ver secção 3.1.2.2), fazemos uso de uma rede que, à partida sabemos que é menos fiável e pior, não temos qualquer controlo sobre a mesma em caso de falha.

Assim, neste caso, a fiabilidade vem:

$$R = (1-S) \times \prod_{i=1}^{n+N+x} (1-a_i) \times \prod_{i=1}^{m+M+x} (1-p_i))$$

Aqui desconhecemos a probabilidade de falha de ligações na rede pública porque desconhecemos o número de equipamentos activos e passivos que separam os nossos clientes DHCP do nosso servidor DHCP.
O simples facto de desconhecermos os dois factores que surgem neste cenário, faz logo com que seja expectável uma muito menor fiabilidade que nos cenários de rede local. O facto de os desconhecermos nem é o mais grave, pois pode-se contratar ligações com SLA's (Service Level Agreements). O grande problema é mesmo o facto de existirem estes factores que contribuem para aumentar a probabilidade de falha do serviço.

3.1.3.2 Caso particular de um operador público de telecomunicações

Um caso particular da utilização do serviço DHCP em redes alargadas é o caso da TV Cabo Portugal (operador de serviços de banda larga por cabo, com posição dominante no mercado através do serviço NetCabo).

Este caso é particularmente interessante pelo facto do número de clientes DHCP ser elevado face aos exemplos anteriores. Por isto, torna-se o caso mais crítico (ver Figura 10).

Mais crítico se torna, quando do serviço DHCP depende também a rede privada de controlo e monitorização. Assim, todo o normal funcionamento do serviço de Internet depende da fiabilidade deste serviço.

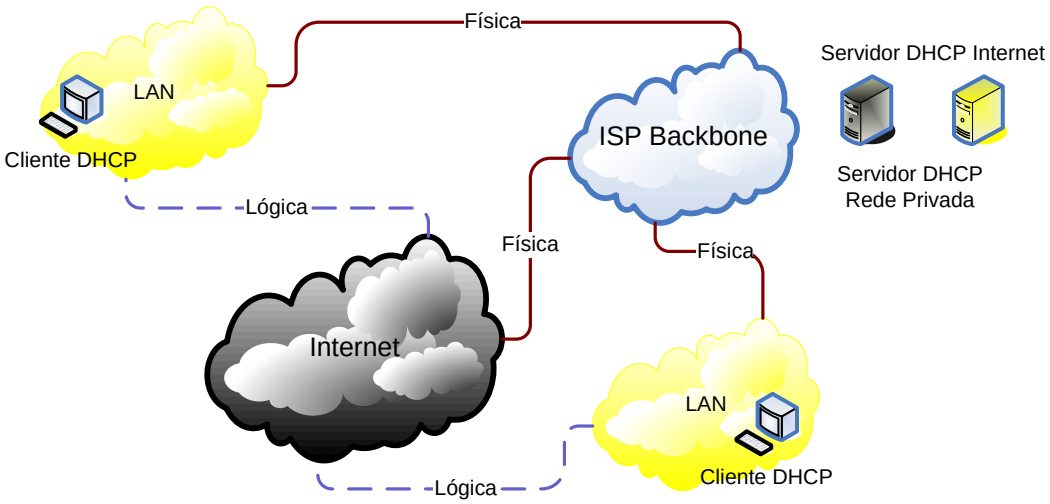

Figura 10: Cenário exemplo de um ISP-BL

A Figura 10 ilustra a dependência que o serviço Internet do operador tem do serviço DHCP.

O objectivo do operador é fornecer o serviço de acesso à Internet aos seus clientes.

Quando um dos seus clientes pretende aceder à Internet liga um equipamento terminal de cliente (neste caso particular, um modem de cabo) à rede do operador. Este processo desenrola-se através de duas fases. Na primeira fase, o equipamento vai tentar obter ligação à rede privada do operador e na segunda fase vai tentar obter/oferecer ligação à Internet.

Na fase inicial, compete ao equipamento terminal negociar uma ligação de nível 2 (neste caso DOCSIS) para poder transmitir e receber dados naquela rede.

Após ter obtido a capacidade de comunicar naquele meio, o equipamento terminal procura obter conectividade de nível 3, i.e. procura obter um endereço IP através um cliente DHCP residente. Aqui surge o primeiro ponto de falha neste cenário associado ao serviço DHCP.
Na ausência do serviço DHCP que gere a rede privada do operador, o cliente não obtém de imediato o acesso ao serviço que contratou (serviço de acesso à Internet).

Mas caso tal não aconteça, e o equipamento terminal (através do seu cliente DHCP residente) consiga obter acesso à rede privada do operador e passar à segunda fase, surge então um novo ponto de falha associado ao serviço DHCP. Este ponto de falha surge quando o equipamento tenta obter ligação à Internet através de um novo pedido de endereçamento IP, não ao servidor que gere o endereçamento da rede privada, mas sim ao servidor que gere a gama de endereços disponíveis para atribuir ligação à rede pública.

Do ponto de vista do operador e da sua relação com o serviço DHCP, podemos afirmar que temos um cenário semelhante ao da Figura 9 onde a rede Internet é agora a rede de interligação (WAN) do operador.

Podemos concluir então que, neste caso, a fiabilidade é semelhante ao cenário de rede privada (ver secção 3.1.2.2) com a agravante de controlarmos duas redes distintas com servidores distintos. Como basta um servidor falhar para todo o serviço falhar temos:

$$R = \prod_{i=1}^{2}(1-S_i) \times \prod_{i=1}^{n+N}(1-a_i) \times \prod_{i=1}^{m+M}(1-p_i))$$

No fundo, um operador de serviço Internet não é mais do que uma entidade que recorre a uma rede alargada para interligar as suas diversas redes locais, com a agravante de fazer uso dessa mesma rede para fornecer acesso à rede Internet obrigando à gestão de duas gamas de endereçamento distintas e para diferentes fins.

3.2 Cenários com redundância

Para aumentar a fiabilidade deste serviço, outros cenários foram experimentados.

Qualquer um dos pontos de falha anteriormente referidos é passível de sofrer medidas de redundância para aumentar a fiabilidade. Além disso outras técnicas (configuração protocolar do serviço, arquitecturas alternativas) são usadas para aumentar/disfarçar a fiabilidade do serviço.

Estes cenários são alvo de estudo e de análise de fiabilidade nesta secção.

3.2.1 Análise de fiabilidade

Na análise seguinte, prova-se que a fiabilidade *R* de um sistema de redundância total de dois elementos (vulgo paralelo) é:

$$R = R_a + R_b - R_a \times R_b$$

Demonstração:

A probabilidade de falha de um sistema em paralelo é igual à probabilidade de falharem os dois elementos em simultâneo. Assim temos:

$$\text{(g)} \qquad P_f(a \cap b) = P_f(a) \times P_f(b)$$

pois a falha de qualquer um deles não depende da falha do outro. Como a fiabilidade é a probabilidade de não falhar (ver secção 3.1.1) donde surge,

$$\text{(h)} \qquad P_f = 1 - R$$

Substituindo (g) em (h) vem o seguinte desenvolvimento,

$$1 - R = P_f(a) \times P_f(b)$$

$$R = 1 - (1 - R_a) \times (1 - R_b)$$

$$R = 1 - (1 - R_b - R_a + R_a \times R_b)$$

ficando por fim,

$$R = R_a + R_b - R_a \times R_b$$

Sendo esta a base de trabalho para cenários com redundância.

3.2.2 Redundância de servidores

Antes da existência do RFC2131, a redundância de servidores era assegurada, pela na introdução um segundo servidor a atribuir endereços numa gama de endereçamento que não se sobrepusesse à original (9). A grande desvantagem era a necessidade de reservar duas gamas de endereçamento para obter a redundância desejada, o que se verificava ser um desperdício.

Com a chegada do RFC2131 ficou contemplada a existência de um segundo servidor DHCP a distribuir endereços na mesma gama de endereçamento (ainda em desenvolvimento (draft)).

A ideia aqui é que o servidor secundário observe o funcionamento do servidor primário e sincronize com ele a informação da atribuição de endereços (endereços e concessões (leases)). Para evitar a possibilidade de duplicação de endereços, o servidor secundário possui um conjunto (pool) de endereços que utiliza, caso o servidor primário falhe.

Para que este sistema funcione, é necessário que a sincronização entre servidores seja perfeita de modo a que, qualquer um deles seja capaz de renovar a concessão de um cliente em qualquer instante.

O mecanismo de sincronização recorre a um sistema de mensagens que são transmitidas entre os dois servidores, contendo a informação das concessões. As mensagens utilizadas são de três tipos:

- Add – mensagem enviada quando ocorre a atribuição de uma nova concessão
- Update – mensagem enviada sempre que ocorre uma alteração na concessão (e.g. renovação da concessão)
- Delete – mensagem enviada sempre que ocorre o fim de uma concessão

Em qualquer caso, as actualizações ocorrem sempre após o processo de negociação com o cliente em causa tiver terminado (lazy updates).

Este protocolo ainda em desenvolvimento prevê a retoma de serviço do servidor principal após falha (8). Para tal, o servidor principal tem de iniciar um processo de negociação com o secundário de modo a assumir novamente o controlo da situação. Novamente três mensagens são necessárias: 1) pedido de tomada de controlo, 2) tomada de controlo iniciada e 3) tomada de controlo terminada.

É esta a parte (processo de negociação com o secundário) que falta desenvolver, pelo que ainda não existem implementações para produção deste protocolo. Por isto assume-se que até à data não existem soluções de redundância de servidores que não obriguem à duplicação das gamas de endereçamento.

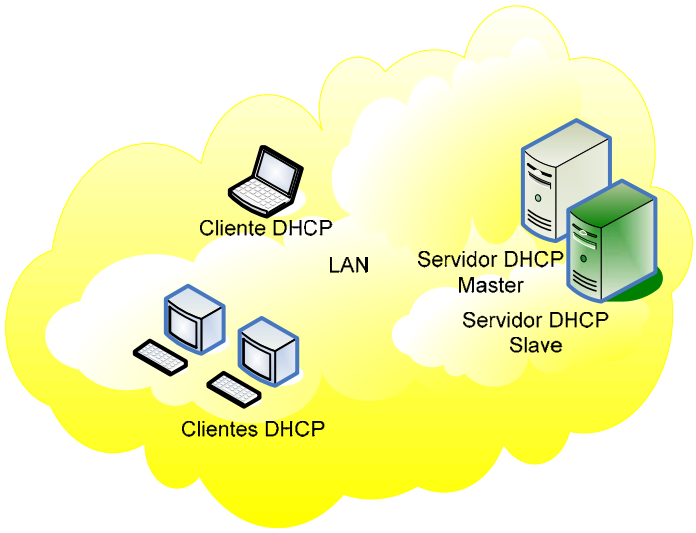

Figura 11: Redundância de Servidores

A redundância de servidores podia-se aplicar em qualquer dos cenários anteriormente analisados pois só sai alterada a probabilidade de falha do servidor S. Assim para o caso mais simples (Figura 11) a fiabilidade do servidor fica equivalente à fiabilidade de dois equipamentos em paralelo (ver secção 3.2.1):

$$(1-S) \rightarrow (1-S_1)+(1-S_2)-(1-S_1)\times(1-S_2)$$

Ficando a fiabilidade total do sistema:

$$R = \left[(1-S_1)+(1-S_2)-(1-S_1)\times(1-S_2)\right]\times\prod_{1}^{n}(1-a)\times\prod_{1}^{m}(1-p)$$

Apesar de tudo, esta solução implicaria sempre uma duplicação da gama de endereçamento e só intervém no problema da falha do servidor, mantendo-se os problemas criados nos cenários de redes distribuídas geograficamente.

A análise teria as mesmas consequências considerando redundância nos equipamentos. Interessa relembrar que todo o tipo de redundância considerado até agora tem um custo que se pode tornar elevado se pretendermos uma fiabilidade elevada.

3.2.3 Configuração do serviço

Uma solução interessante, pelo facto de possuir um custo nulo de implementação, é alterar a configuração do serviço. O parâmetro que contribui para melhorar a performance do serviço é o tempo de concessão (lease time) atribuído aos endereços.

O protocolo permite que especifiquemos tempos de concessão elevados (dias ou semanas). Com esta solução pretende-se que as máquinas que estão ligadas à rede não percam os parâmetros de rede, pois demoram tanto tempo a solicitar a renovação dos mesmos ao servidor, que entretanto a falha é solucionada.

No entanto, no período de falha nenhuma máquina nova conseguiria ligar-se à rede e o servidor não teria informação das máquinas que tinham saído da rede, mantendo os seus endereços atribuídos a estas (embora tal não constituísse problema).

Resumindo, perco toda a flexibilidade e vantagens pelas quais se optou por este serviço (DHCP) inicialmente.

3.2.4 Gestão descentralizada

A solução (Figura 12), que resolve os problemas de conectividade (pois assume-se que são nulos nas redes locais em comparação com a Internet) tem o severo inconveniente de ter um custo de implementação elevado (o que poderia compensar ao resolver o problema da indisponibilidade), que implicaria a perda total da flexibilidade que uma gestão centralizada oferece e que é uma grande vantagem oferecida pelo serviço.

Além disso uma gestão local, também implica um custo muito superior em recursos humanos para além do custo da implementação. Não podemos esquecer que, nos cenários anteriores, tínhamos um servidor para nos preocupar, enquanto neste caso passaríamos a ter tantos servidores como redes locais. O que faríamos? Colocávamos redundância em todos eles?

Por esta razão é uma opção raramente tomada pelos responsáveis dos sistemas de informação das empresas.

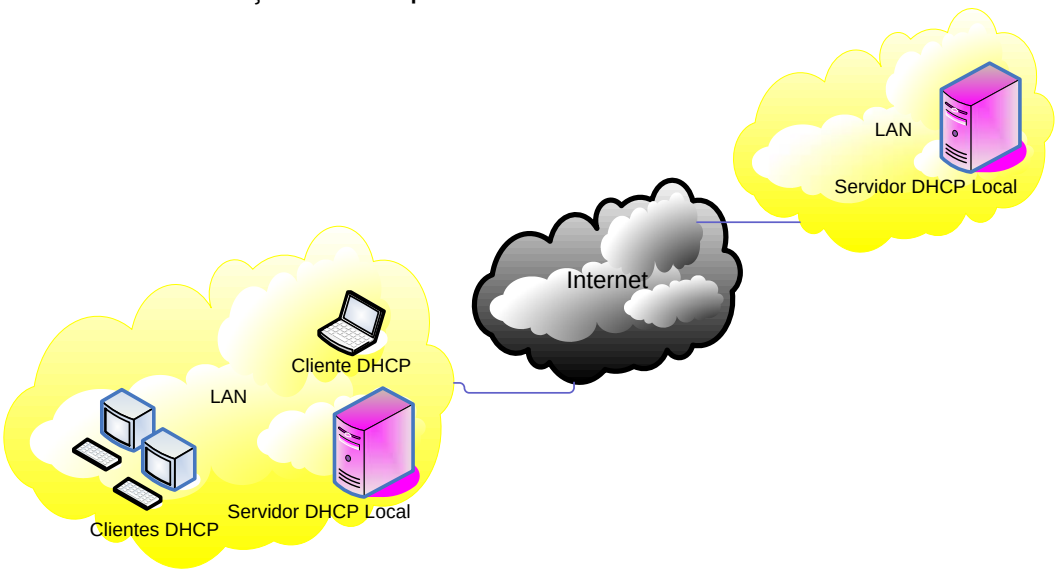

Figura 12: Distribuição do serviço utilizando servidores locais

3.3 Análise Comparativa

Nesta secção fazemos uma análise comparativa entre os vários cenários previamente discutidos e respectivas soluções de redundância. A Tabela 5 mostra um resumo dos vários cenários que são detalhados nas secções seguintes.

3.3.1 Cenários de Utilização

O cenário da rede local, foi considerado neste capítulo por motivos de enquadramento, pois não se verifica neste, o segundo constrangimento referido em 1.2, i.e. não podemos considerar a existência de uma gestão centralizada de clientes remotos, mas sim de uma gestão, também ela centralizada, mas de clientes locais. Neste cenário, a simples redundância local (recorrendo a um servidor secundário, como previsto no protocolo) é suficiente para garantir a continuidade do serviço.

Quando passamos de uma rede local para uma rede logicamente alargada, mas não geograficamente distribuída, já começamos a viver o problema causado pela gestão centralizada. No entanto, sendo as redes geograficamente próximas e provavelmente, com todas as ligações controladas pelos administradores da rede, consegue-se com facilidade garantir uma elevada fiabilidade do serviço.

Analisando o cenário considerado no âmbito desta dissertação (redes alargadas de redes locais distribuídas geograficamente), verifica-se que, a utilização de interligações não proprietárias ou o recurso à Internet podem representar um severo constrangimento para o bom funcionamento de toda a rede. Estando os clientes DHCP de uma determinada rede local dependentes de um servidor localizado central e remotamente, qualquer falha da interligação entre a rede local e a rede remota (perda de conectividade remota) provocará a perda da conectividade local. É neste cenário que a fiabilidade do serviço baixa drasticamente e é neste cenário que concentramos a utilidade da solução de redundância distribuída.

Um caso particular deste cenário é o do ISP de banda larga, onde por motivos de gestão, o serviço ao cliente final depende duplamente de um bom funcionamento do serviço DHCP, sendo por essa razão, ainda mais crítico.

Tabela 5: Quadro Resumo de Cenários de Utilização e Soluções de Redundância

Cenário de Utilização	Vantagens	Desvantagens	Fiabilidade do Serviço
Rede Local	Gestão local. Probabilidades de falha conhecidas.		Elevada, determinística
Rede Local – Segmentos Diferentes	Gestão local. Probabilidades de falha conhecidas.	Aumenta a probabilidade de falha de conectividade	Elevada, determinística
Rede Alargada – Empresa	Gestão centralizada	Existência de um ponto falha fora do nosso controlo (negociado ou não) fiabilidade	Mais baixa que em redes locais
Rede Alargada – Caso particular	Gestão centralizada. Backbone é controlado pela entidade	Dupla utilização do serviço	Mais baixa que redes locais, mas superior ao cenário de recurso à Internet
Redundância de Servidores	Aumenta a fiabilidade no outro extremo do sistema	Requer dupla gama de endereçamento. Não resolve o grave problema da conectividade	Fiabilidade na componente do servidor
Redundância de Ligações	Aumenta a fiabilidade da conectividade	Caro. Desperdício de recursos poderá não compensar o incremento de fiabilidade	Aumenta a fiabilidade com custos elevados
Configuração do Serviço	Custo nulo. Resolve problema de conectividade e de servidor	Só resolve problema a máquinas já ligadas à rede. Provoca problemas de gestão caso os parâmetros sejam alterados	Fiável mas inútil em redes de elevada mobilidade
Gestão Descentralizada	Problema de conectividade fica reduzido à rede local	Perde-se a flexibilidade de uma gestão centralizada. Custos elevadíssimos em equipamento e recursos humanos	Aumenta a fiabilidade de uma rede alargada para ficar equivalente a rede local

3.3.2 Soluções de Redundância

A redundância de servidores, conforme referido anteriormente, serve apenas para aumentar a robustez de funcionamento do serviço a nível central. Não trás qualquer mais-valia para o problema de conectividade com a agravante de, até ao momento, ainda não existirem soluções que evitem uma dupla gama de endereçamento (claro desperdício). Assim, esta solução torna-se apenas algo interessante para cenários de rede local, havendo disponibilidade para desperdiçar gamas de endereçamento.

Recorrendo à redundância de ligações, já começamos a atacar o problema de que é alvo a nossa análise. Colocando redundância nas saídas WAN das redes locais conseguimos aumentar a fiabilidade do serviço no cenário das redes alargadas. No entanto, esta solução requer um investimento que poderá se tornar insustentável dependendo do grau de fiabilidade pretendida.

3.3.3 Outras soluções

Outras formas existem para contornar os constrangimentos. O protocolo permite que o período de tempo ao fim do qual o cliente tem de contactar o servidor (tempo de concessão) possa ser aumentado até valores muito elevados. Desta forma, é possível aumentar este tempo (através da configuração do serviço) para que seja sempre superior ao tempo de uma falha qualquer (conectividade ou servidor). Mesmo existindo uma falha, o cliente não terá necessidade contactar o servidor e não perderá o endereçamento local. Esta solução tão simples que chega a parecer espectacular, trás agregada a si um inconveniente: só funciona para clientes que já estejam ligados à rede. No cenário de mobilidade que vivemos nos dias de hoje, onde os computadores são portáteis e as pessoas se deslocam entre sectores e locais das empresas, esta solução não é útil, pois os clientes novos ficam sempre impossibilitados de se ligar à rede na presença de uma falha.

Uma outra solução passaria por descentralizar o serviço. Todas as redes locais ficariam com o seu próprio servidor, e a fiabilidade do serviço seria equivalente ao de uma só rede local. Mas perder-se-ia uma das vantagens que nos levou a optar inicialmente pela utilização do DHCP: a gestão centralizada. Provavelmente necessitaríamos de aumentar os recursos afectos à gestão do serviço para poder gerir todos os servidores de forma independente.

No capítulo seguinte, apresenta-se uma solução alternativa: a redundância distribuída.

4 Redundância distribuída

O uso do protocolo DHCP simplifica em todos os aspectos o trabalho do administrador de rede. Gere centralmente a atribuição de endereços e as máquinas autoconfiguram-se. No entanto, o bom funcionamento deste serviço requer que o servidor DHCP esteja disponível durante todo o tempo de operação da rede.

Este problema assume proporções maiores, se não pensarmos exclusivamente nas máquinas que já estão ligadas à rede, pois estas ainda têm o tempo de concessão (lease time) de margem. Se pensarmos que as redes actuais assumem-se cada vez mais como redes de máquinas portáteis e/ou sem fios (wireless), tendo por isso, uma maior mobilidade, então teríamos um conjunto de utilizadores que ficariam desde logo impossibilitados de se ligar à rede na ausência do servidor DHCP.

O cenário especialmente considerado para esta análise é o pior possível, onde existe uma rede pública a ligar as diversas redes locais e não existe qualquer redundância (servidores, ligações, equipamentos). O cenário encontra-se ilustrado na Figura 13.

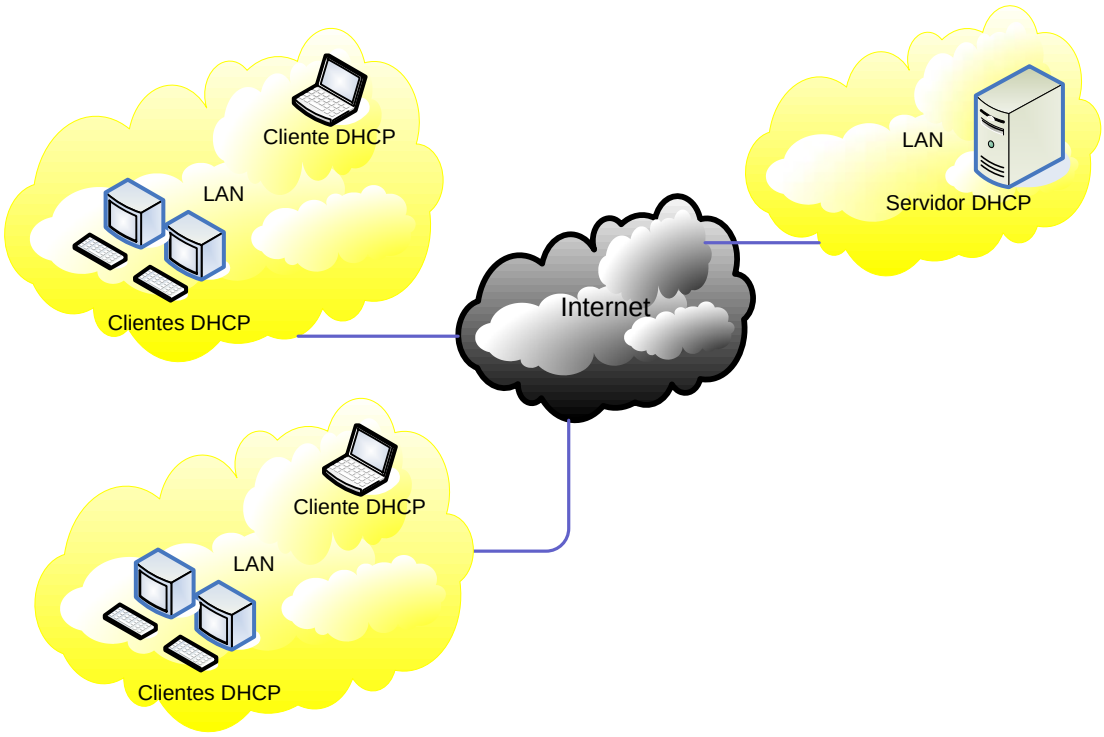

Figura 13: Cenário de Análise

4.1 Proposta de Solução

Neste trabalho, é proposta uma solução de custo reduzido que não requer gestão, que colmata as lacunas sentidas pelos administradores que se deparam com a generalidade dos cenários anteriormente descritos.

O objectivo da redundância distribuída, é dotar os diversos sítios (sites) com um sistema inteligente que obedeça às seguintes premissas:

1. Qualquer falha de conectividade ou de servidor, deve passar incólume aos clientes do serviço
2. Qualquer mudança no estado de configuração do serviço após falha, deve ser transparente para o servidor quando retomar o normal funcionamento

Em resumo, quer o cliente, quer o servidor não devem ter conhecimento de que a falha ocorreu.

Para tal, pretende-se dotar os pontos críticos deste tipo de topologias (entenda-se, saídas para a rede alargada das diversas redes locais), de um sistema que:

a) Em normal funcionamento, consiga ter em determinado instante uma "fotografia" do estado do serviço, "fotografia" essa, obtida por aprendizagem.
b) Seja capaz de detectar as falhas de normal funcionamento
c) Seja capaz de se substituir ao servidor do ponto de vista dos clientes
d) Seja capaz de informar o servidor das modificações no estado do serviço, após este voltar ao estado de normal funcionamento
e) Não necessite de configuração com parâmetros de rede

Este último ponto é especialmente importante pois pretende-se uma máquina autónoma que não exija gestão e.g. em caso de mudanças na configuração ou mudança de gama de endereçamento. Desta forma, permite-nos a sua colocação em qualquer rede sem necessidade de configuração.

Para distinguir o servidor DHCP central do servidor DHCP redundante distribuído, designaremos daqui para frente o primeiro como servidor e o segundo como iDHCP.

4.2 Arquitectura da solução

A arquitectura proposta na Figura 14 consiste em introduzir nos pontos críticos das sub-redes (provavelmente junto dos encaminhadores (routers) para a Internet) uma máquina que, em modo promíscuo, "oiça" todas as conversações de DHCP, mantenha actualizada uma imagem do estado da rede (parâmetros de rede e temporizadores, por máquina), detecte situações de falha e se substitua ao servidor central durante o período em que este se encontre indisponível.

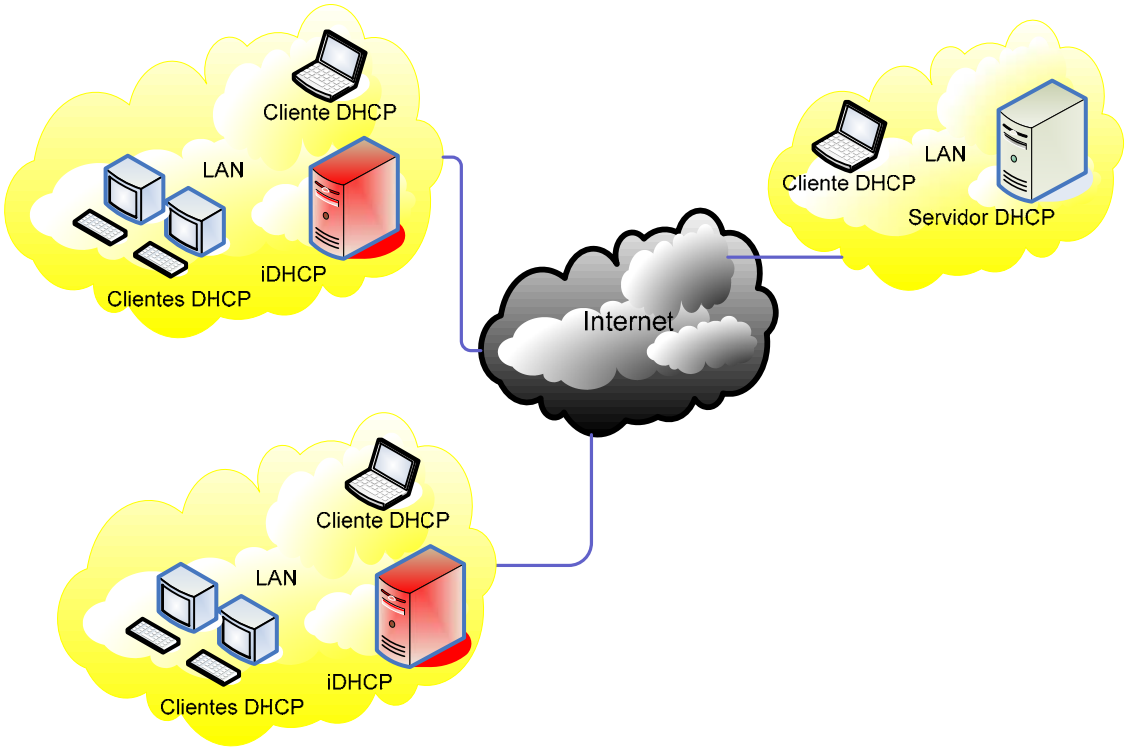

Figura 14: Arquitectura

4.2.1 Localização na rede

Para que esta máquina consiga capturar todo o tráfego da rede local tem de ser colocada em pontos da rede onde tal seja possível.

Nesta análise assume-se que as redes actuais são todas comutadas, i.e. recorrem a comutadores (switches) para concentrar as ligações físicas entre os diversos equipamentos da rede.

De uma forma geral (caso em que o comutador não possui uma porta promíscua) o iDHCP deve ser colocado entre o encaminhador (router) e o comutador (switch) da rede. Neste caso é possível recorrer a uma placa de rede que, em caso de falha da máquina, faz automaticamente a ponte (bypass) do circuito para a rede local (Figura 15), com o fim de evitarmos um novo ponto de falha.

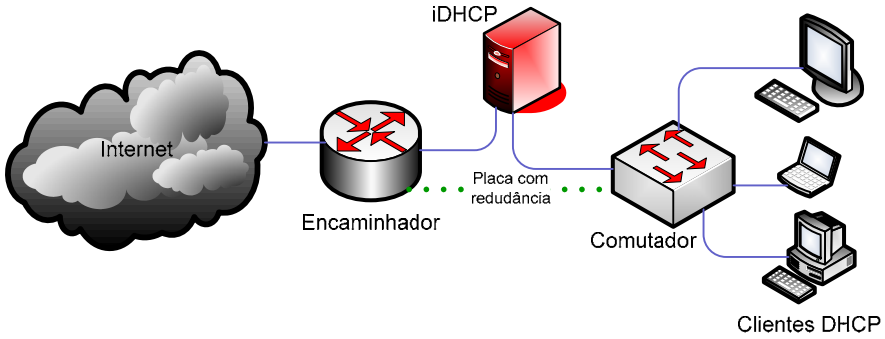

Figura 15: Solução com bypass

No caso do comutador de rede possuir uma porta promíscua, podemos ligar o iDHCP a uma porta do comutador em modo promíscuo (Figura 16). Neste caso não existe a problemática de um ponto adicional de falha pois estamos a utilizar um equipamento já existente na rede, mas para outros fins.

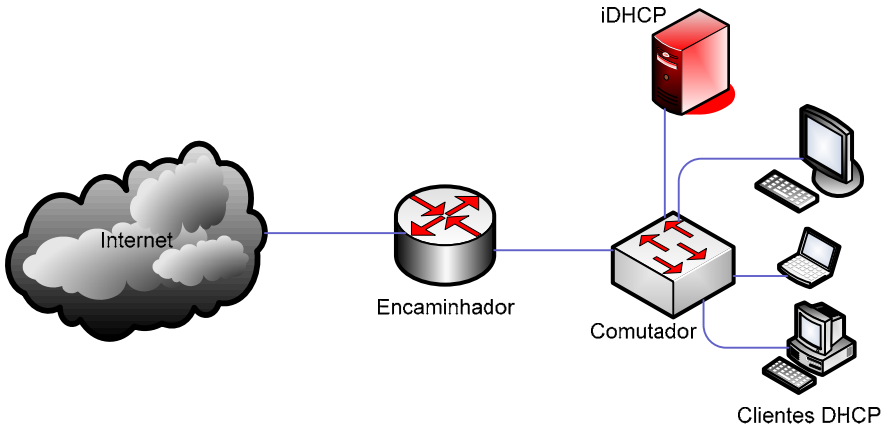

Figura 16: Solução com porta promíscua

Ainda como solução alternativa à placa com redundância (através do bypass) colocar-se-ia um concentrador (hub) neste segmento, com a desvantagem de, neste caso, o concentrador se tornar um ponto adicional de falha (Figura 17).

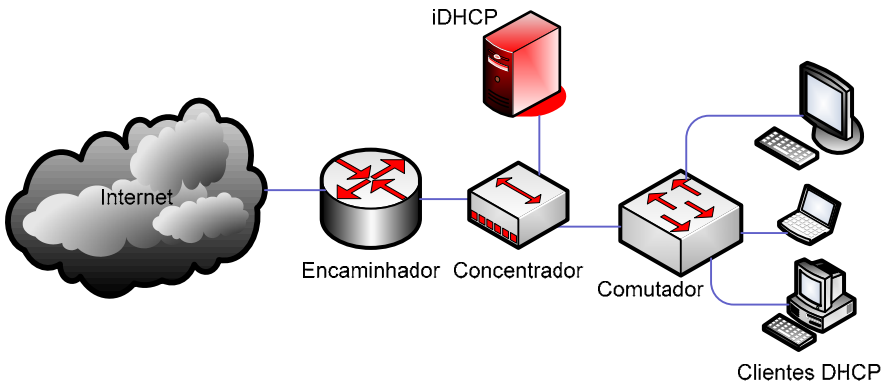

Figura 17: Solução com concentrador

É fácil constatar que a solução com comutador (switch) com porta promíscua é a mais transparente para a rede onde vai ser colocado o iDHCP. Esta solução não adiciona qualquer ponto de falha e não exige a utilização de qualquer tipo de equipamento (hardware) adicional.

4.2.2 Diagrama de estados

A detecção de falhas é implementada pelo sistema de redundância distribuída. Este possui um algoritmo com um certo grau de capacidade de aprendizagem que capture todo o tráfego DHCP e deste modo, registe, aprenda e reaja a modificações no estado do serviço de modo a cumprir as premissas anteriormente enumeradas.

A Figura 18 ilustra o diagrama de estados que se pretende:

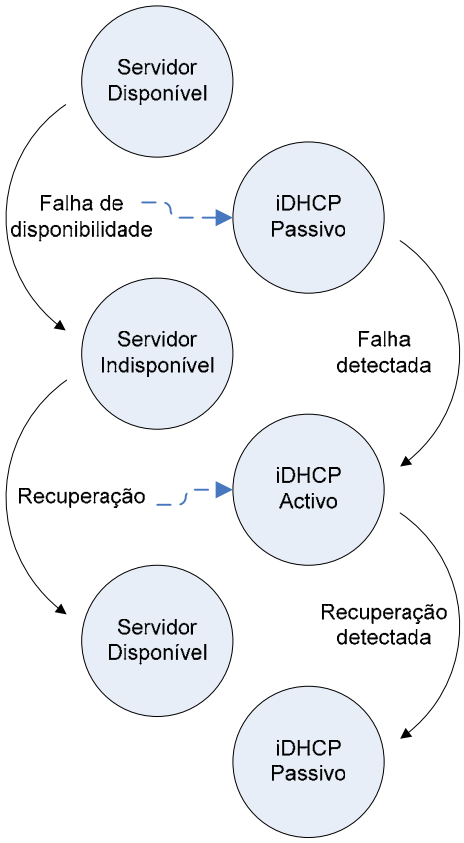

Figura 18: Diagrama de estados do sistema de redundância distribuída

Inicialmente temos o estado normal de funcionamento. O servidor encontra-se activo e embora o iDHCP se encontre a monitorizar o tráfego, considera-se que se encontra no estado passivo pois não intervém no estado do serviço.

Para despoletar uma mudança de estado é necessário que o iDHCP detecte que o servidor deixou de responder aos pedidos dos clientes (ver secção 4.3.1). Com esta ocorrência o iDHCP passa ao estado activo, substituindo-se ao servidor e respondendo aos pedidos dos clientes (ver secção 4.3.2).

O iDHCP mantém a monitorização ao tráfego enquanto responde aos pedidos dos clientes. Quando o iDHCP detecta que o servidor retomou a actividade (ver secção 4.3.3), devolve o controlo do serviço ao mesmo voltando ao estado passivo. Este processo desenrola-se em ciclo infinito.

4.2.3 Algoritmos possíveis

A implementação da solução do sistema de redundância distribuída pode ser feita com distintas abordagens. Uma de mais alto nível (implementação mais imediata mas menos robusta e dependente de pacotes de software de terceiros), recorrendo a software já existente e outra de mais baixo nível recorrendo a uma implementação directa de todas as fases do algoritmo. Entenda-se que esta distinção nada tem a ver com a possibilidade de utilizarmos ou não uma linguagem de alto ou baixo nível. Em qualquer dos casos a linguagem utilizada é sempre de muito alto nível (PERL). A distinção existe porque num caso é necessário implementar todas as fases do algoritmo e noutro caso não.

Independentemente da abordagem, a solução tem sempre de contemplar as seguintes fases do processo (ver Figura 19):

1. Captura de pacotes da rede
2. Filtragem de pacotes DHCP
3. Descodificação dos campos e das opções do pacote DHCP
4. Decisão e actuação
5. Construção dos pacotes DHCP
6. Construção do pacote de rede final
7. Envio dos pacotes para a rede

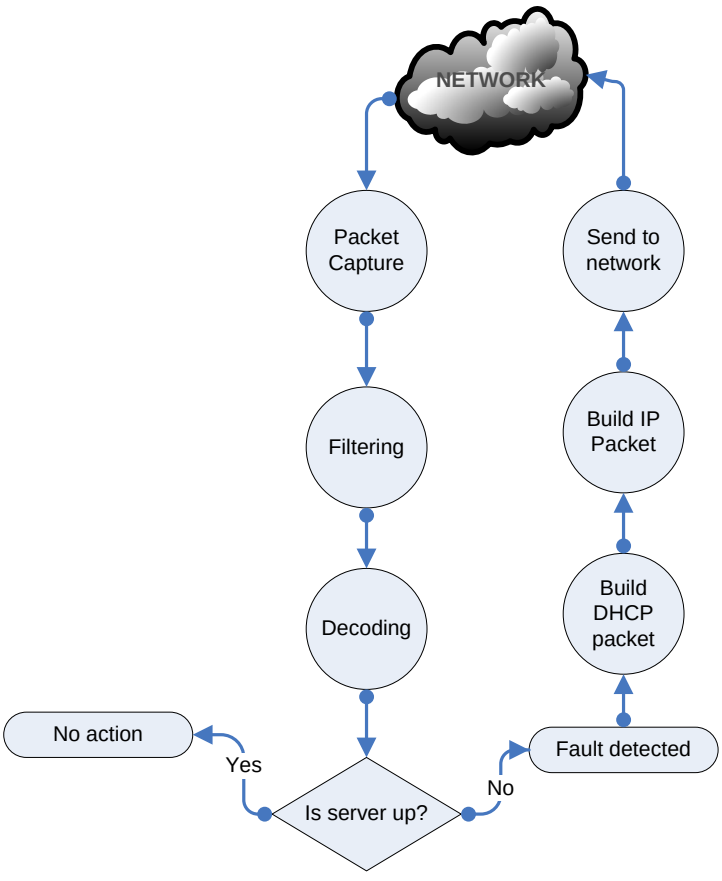

Figura 19: Algoritmo genérico

4.2.3.1 Abordagem de mais alto nível

A abordagem de mais alto nível é a apresentada na Figura 20. Este sistema faz uso de ferramentas já existentes (tcpdump e dhcpdump) para efectuar o processo de captura e filtragem. No entanto não permitem a operação inversa, sendo necessária a sua implementação.

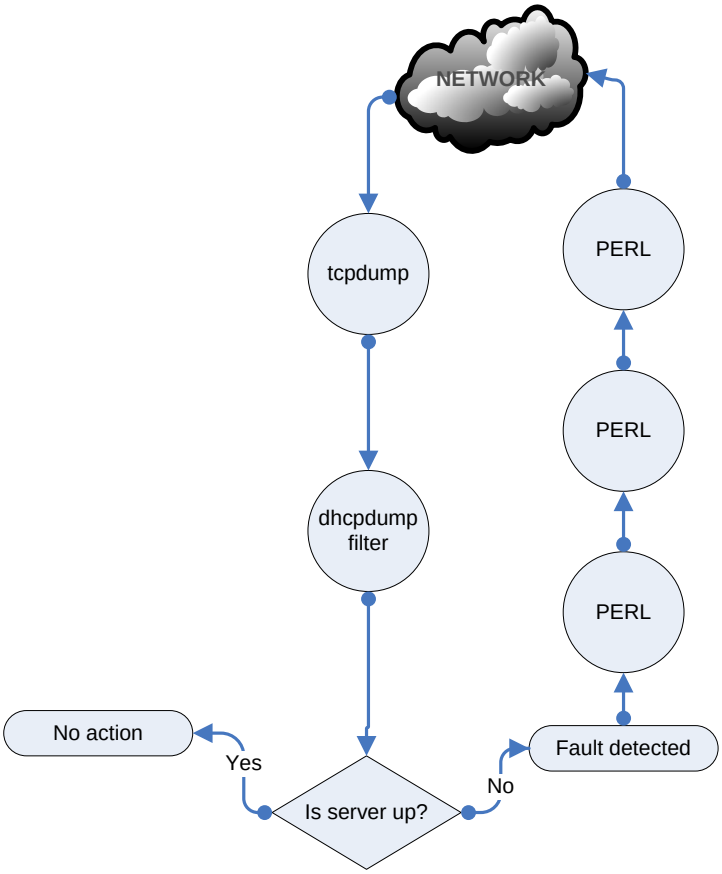

Figura 20: Abordagem de alto nível

Estes pacotes de software são meramente indicativos pelo que existirão outros que sirvam o mesmo fim, todavia estes foram os escolhidos e devidamente testados para a implementação do algoritmo.

4.2.3.2 Abordagem de baixo nível

Uma alternativa passa por desenvolver em todas as fases o código necessário à implementação do sistema (Figura 21).

A necessidade imposta pela parte de construção do pacote, implica um conhecimento profundo de todos os campos de um pacote DHCP. Esta metodologia faz com que tenhamos mais controlo, mesmo no processo de captura, permitindo tomar opções que sejam mais adequadas para o nosso objectivo.

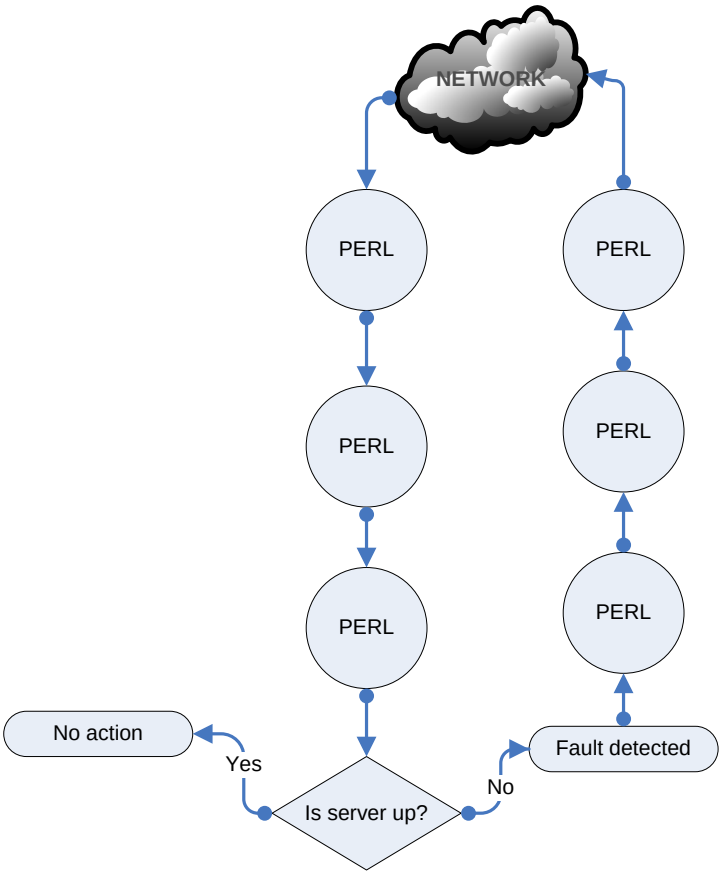

Figura 21: Abordagem de baixo nível

Considera-se também, que esta solução seja a mais elegante a nível de desenvolvimento, pois deixa-nos imunes a eventuais falhas existentes nos pacotes de software sugeridos pela hipótese anterior.

4.2.3.3 Escolha da linguagem Perl

Optou-se pela escolha de uma linguagem "scríptica" que recorre a um interpretador em detrimento de uma outra qualquer linguagem de referência como o C porque, além de estar perfeitamente orientada para a programação em comunicação, o Perl é uma ferramenta poderosa de análise e tratamento de cadeias de caracteres (aliás construída para isso) que vai perfeitamente de encontro às necessidades de pretende capturar pacotes e analisá-los.

4.3 Métodos

Aprofundaremos nesta secção o funcionamento do pacote de software inteligente, pois é o cerne de toda a questão.

Iremos subdividir a análise em quatro pontos:

- Detecção da falha de disponibilidade do servidor
- Processo de substituição do servidor pelo iDHCP
- Detecção de recuperação de disponibilidade do servidor
- Actualização do estado do servidor após falha

4.3.1 Detecção de falha de disponibilidade

Nesta fase temos sempre de considerar dois tipos de clientes DHCP distintos, por razões de diferença comportamental:

- Clientes novos que entram na rede com o servidor indisponível
- Clientes já com endereço atribuído na rede com necessidade de renovarem as suas concessões recorrendo a um servidor que se encontra indisponível

4.3.1.1 Clientes novos

Observando as conversações, é possível verificar pela análise dos pacotes, que o iDHCP detecta a falta do servidor quando uma mensagem do tipo DHCPDISCOVER fica sem qualquer resposta DHCPOFFER.

Por exigência protocolar, o cliente DHCP tem de preencher o campo "chaddr" (endereço de hardware do cliente) ao construir a mensagem DHCPDISCOVER.

O(s) servidor(es) irá incluir obrigatoriamente este campo na mensagem DHCPOFFER de resposta.

Como este endereço é universalmente único (controlado pelo IANA a nível mundial junto dos fabricantes de hardware), deve ser usado para implementar o controlo das mensagens DHCPDISCOVER sem resposta, por cliente.

Podendo identificar os clientes, temos de escolher um critério para identificar quando a falha ocorre. Como o período de tempo a partir do qual o cliente desiste de esperar pela resposta (timeout) pode ser diferente de cliente para cliente (visto que é um parâmetro configurável), optamos por escolher um critério de contagem.

Assim, considera-se que ocorreu uma falha do servidor central (entenda-se, qualquer falha que impossibilita a comunicação entre cliente e servidor), quando foram detectadas três mensagens DHCPREQUEST sem resposta (ver Figura 22).

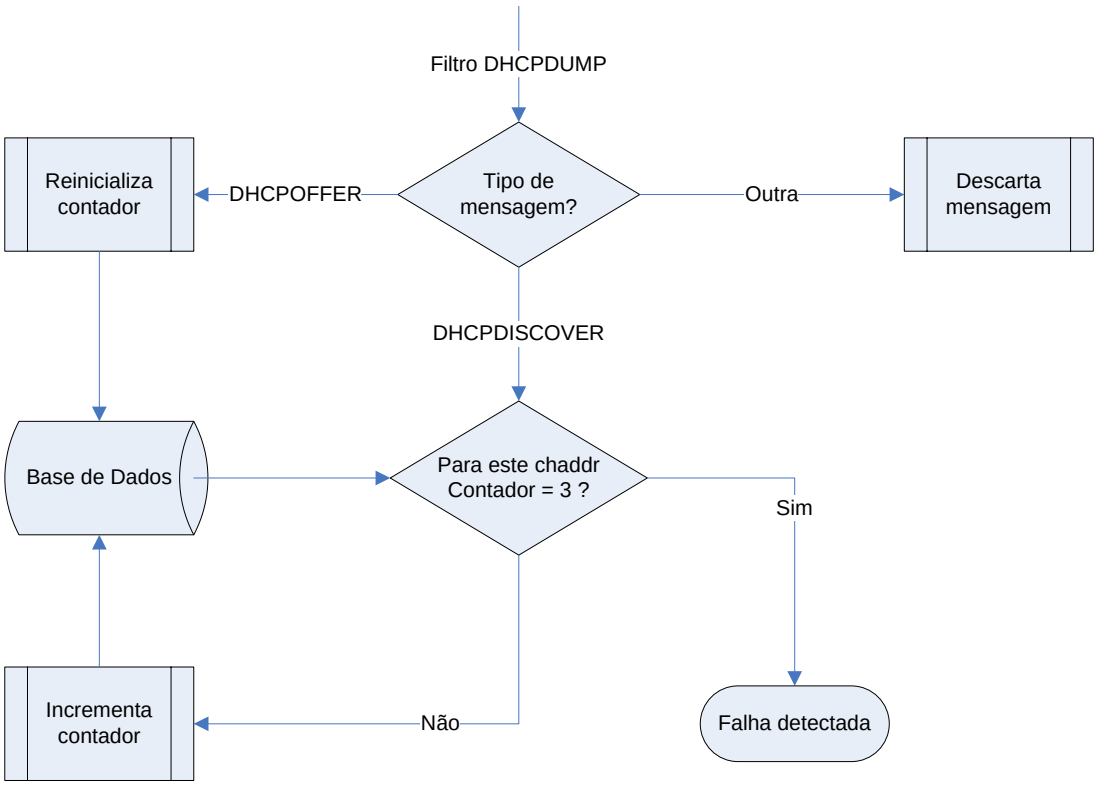

Figura 22: Algoritmo de detecção de falhas I

Uma alternativa seria optar por um algoritmo de aprendizagem do atraso da rede (network lag), mas parece-nos que a relação fiabilidade/complexidade sai prejudicada.

4.3.1.2 Clientes já presentes na sub-rede

Neste caso, a falha será detectada quando o iDHCP se aperceber que alguma mensagem DHCPREQUEST ficou sem o DHCPACK correspondente. O algoritmo será idêntico ao anterior (ver Figura 23).

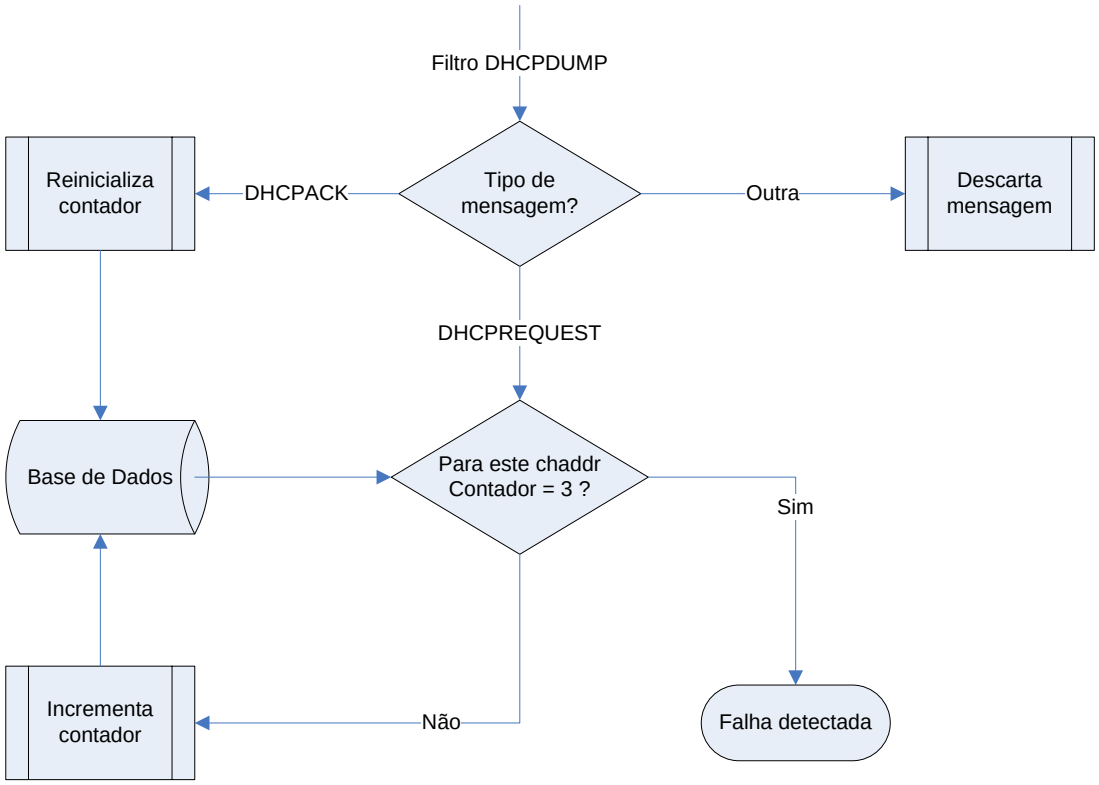

Figura 23: Algoritmo de detecção de falhas II

4.3.2 Substituição ao servidor central

O iDHCP deve construir as mensagens DHCP de substituição para que os clientes não se apercebam da substituição, possibilitando uma "graceful way out" quando o servidor central recuperar.

Compete então ao iDHCP criar mensagens que sejam exactamente iguais (salvo alguns campos que teremos necessidade de modificar para atingirmos o nosso objectivo) às mensagens que seriam criadas pelo servidor central.

Isto levanta várias questões, entre muitas outras:

- Como saber qual o conjunto de endereços que o servidor usa?
- Quais os endereços disponíveis?
- Quais os parâmetros de configuração?

Fica claro, por estas razões, que o iDHCP tem de possuir um mecanismo de aprendizagem do estado da rede e respectivos parâmetros.

4.3.2.1 Mecanismo de aprendizagem

O iDHCP, no estado passivo de observação, deve guardar e manter uma fotografia do estado da rede. O iDHCP tem de ser capaz de:

1 Aprender/conhecer o mapa de endereçamento da sub-rede
2 Aprender/conhecer a gama (scope) de endereçamento da sub-rede
3 Aprender/conhecer os parâmetros específicos da sub-rede

- Mapa de endereçamento

Com fim de criar e manter um mapa de endereçamento, o iDHCP tem de guardar os campos "chaddr" e "yiaddr" de todas as mensagens DHCPACK que forem interceptadas construindo uma tabela como e.g. a Tabela 6.

Tabela 6: Exemplo de Mapa de Endereçamento

Endereço de hardware	Endereço de rede
00:A0:50:AE:07:B6	192.168.4.26
00:D0:C6:22:55:77	192.168.4.7
...	...
...	...
00:A0:50:3C:BD:07	192.168.0.1

- Gama de endereçamento

Com uma simples mensagem capturada é possível através da máscara de sub-rede, determinada qual a gama de endereçamento da sub-rede em causa. A gama deve ser actualizada periodicamente para detectar alterações de configuração no servidor central.

- Parâmetros específicos

  - Endereço IP do servidor
  - Tempo de expiração T1
  - Tempo de expiração T2
  - Tempo de concessão (lease) do IP

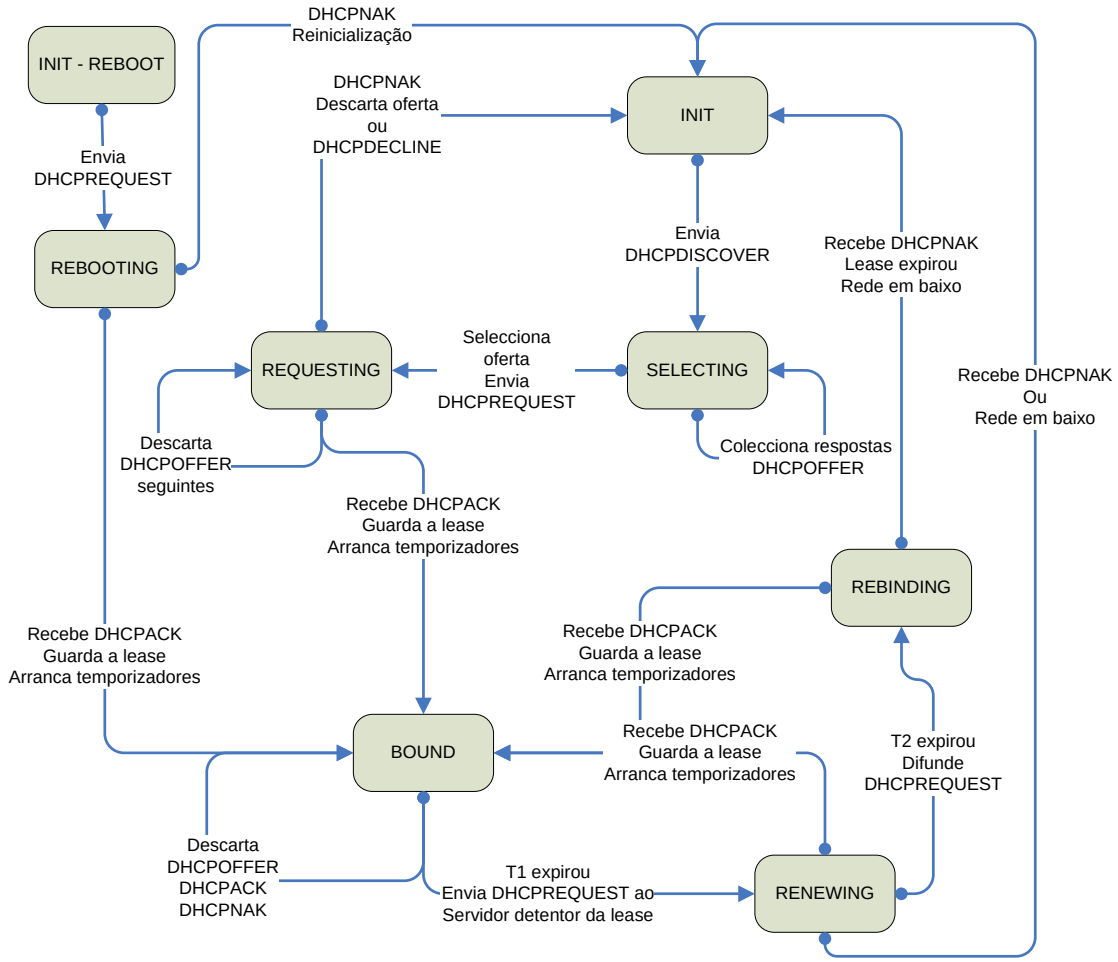

Figura 24: Diagrama de estados de um cliente DHCP

4.3.2.2 Mecanismo de substituição

Para que o iDHCP se substitua ao servidor central temos de analisar o comportamento esperado pelos clientes em cada um dos seus estados (Figura 24). Assim, recordemos o diagrama de estados de um cliente DHCP para melhor percebermos a problemática (ver Figura 24).

- Estado INIT

  Quando se encontra no estado INIT, i.e. quando inicializa o processo de configuração observa-se que o cliente só emite mensagens DHCPDISCOVER. Neste momento o que o cliente pretende, é receber uma mensagem DHCPOFFER de um qualquer servidor da rede com os parâmetros de configuração da sub-rede.

  O que se pretende então, é que o iDHCP construa uma mensagem DHCPOFFER (ver Tabela 7) onde se identifica como sendo o servidor central. Para tal tem de preencher as opções e os campos da mensagem conforme os parâmetros adquiridos durante o processo de aprendizagem.

Aqui é importante que o iDHCP preencha a opção “server identifier” com o endereço IP do servidor central (assume-se que esta máquina tenha IP fixo, como é de esperar), permitindo assim que as mensagens DHCPREQUEST seguintes (mensagens com destino definido após T1 se esgotar) sejam direccionadas ao servidor DHCP central. O IP tem de ser um IP disponível da gama da sub-rede, pois o cliente irá rejeitar a oferta se detectar que já está a ser usado.

- Estado SELECTING e REBOOTING

Neste estado os clientes apenas aguardam respostas dos servidores, não emitindo qualquer mensagem.

- Estado REQUESTING e INIT-REBOOT

Nestes estados o cliente solicita os parâmetros de configuração ao servidor seleccionado. O cliente espera receber uma mensagem DHCPACK (ou DHCPNAK) do servidor em causa.

Aqui, o iDHCP deve construir uma mensagem DHCPACK usando os parâmetros que adquiriu no processo de aprendizagem. Mais uma vez a opção “server identifier” deve ser o endereço IP do servidor central.

- Estado RENEWING

O tempo T1 expirou (50% do tempo da concessão (lease)), o cliente envia uma mensagem DHCPREQUEST para o servidor central (este campo nunca varia, nem com a intervenção do iDHCP).

O iDHCP intercepta e constrói uma mensagem DHCPACK de acordo com os parâmetros que se encontram na mensagem DHCPREQUEST do cliente. O tempo da concessão (lease) é novamente inserido (tempo retirado do mecanismo de aprendizagem) e o cliente mantém-se configurado.

- Estado REBINDING

O iDHCP comporta-se da mesma forma que no estado RENEWING, pois as diferenças ocorrem ao nível do cliente (expiração do tempo T2 e envio das mensagens DHCPREQUEST em difusão).

Nota: Espera-se que nesta solução não se atinja este estado, pois implica que tenha existido uma mensagem DHCPREQUEST sem resposta, o que temos como objectivo evitar.

- Estado BOUND

Neste estado o cliente encontra-se configurado a aguardar que os temporizadores despoletem uma mudança de estado. O sistema encontra-se estável.

Tabela 7: Campos (relevantes) das mensagens enviadas pelo servidor

Campo	DHCPOFFER	DHCPACK	DHCPNAK
op	BOOTREPLY	BOOTREPLY	BOOTREPLY
htype	Retirado do RFC "Assigned Numbers"		
hlen	Comprimento do endereço de HW em octetos		
hops	0	0	0
xid	Identificação de transacção presente na mensagem DHCPDISCOVER do cliente	Identificação de transacção presente na mensagem DHCPREQUEST do cliente	
secs	0	0	0
ciaddr	0	"ciaddr" do DHCPREQUEST ou 0	0
yiaddr	Endereço IP oferecido ao cliente	Endereço IP atribuído ao cliente	0
siaddr	Endereço IP do servidor DHCP	Endereço IP do servidor DHCP	0
flags	"flags" idênticas às recebidas na mensagem DHCPDISCOVER do cliente	"flags" idênticas às recebidas na mensagem DHCPREQUEST do cliente	"flags" idênticas às recebidas na mensagem DHCPREQUEST do cliente
giaddr	"giaddr" presente na mensagem DHCPDISCOVER do cliente	"giaddr" presente na mensagem DHCPREQUEST do cliente	"giaddr" presente na mensagem DHCPREQUEST do cliente
chaddr	"chaddr" presente na mensagem DHCPDISCOVER do cliente	"chaddr" presente na mensagem DHCPREQUEST do cliente	"chaddr" presente na mensagem DHCPREQUEST do cliente
sname	Nome do servidor ou opções	Nome do servidor ou opções	(vazio)
file	Ficheiro de arranque ou opções	Ficheiro de arranque ou opções	(vazio)
Opção			
Requested IP address	(vazio)	(vazio)	(vazio)
IP address concessão (lease) time	Tempo em segundos 4 octetos 32-bit inteiro sem sinal	DHCPREQUEST (tempo em segundos) DHCPINFORM (vazio)	(vazio)
DHCP message type	DHCPOFFER(2)	DHCPACK(5)	DHCPNAK(6)
Message	Deve ser incluída uma mensagem	Deve ser incluída uma mensagem	Deve ser incluída uma mensagem
Client identifier	(vazio)	(vazio)	(opcional) Par "htype"/"chaddr"
Server identifier	Endereço IP do servidor	Endereço IP do servidor	Endereço IP do servidor
OUTRAS	(opcional)	(opcional)	(vazio)

4.3.3 Detecção de recuperação de disponibilidade

Como pode o iDHCP detectar que o servidor central está de novo disponível?

1. Fazendo polling ao servidor a partir do momento que é detectada uma falha
2. Analisando as conversações DHCP

4.3.3.1 Polling

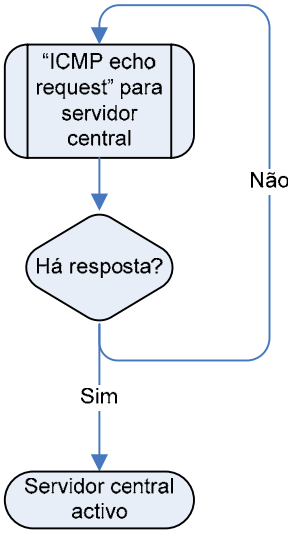

Figura 25: Polling

Esta solução (ver Figura 25) vai um pouco contra a definição de sistema inteligente, mas não deve ser descartada, pois não se provou ser a menos eficiente (ver secção 4.5.3). Uma possível implementação desta solução seria enviar um falso DHCPDISCOVER ao servidor e capturar a resposta. Há no entanto um cuidado a ter com este tipo de situações: as mensagens enviadas pelo iDHCP ao servidor, podem despoletar reacções do próprio iDHCP, pois é essa a sua essência de funcionamento. Por esta razão este assunto permanece em aberto.
Outra possibilidade mais tradicional passa enviar uma mensagem "ICMP echo request" e verificar a resposta. Esta solução tem uma desvantagem. O iDHCP desconhece a razão pela qual não consegue contactar o servidor. No caso da razão de falha do servidor, ser da responsabilidade do serviço e não da máquina, podemos ter a máquina a responder aos pedidos ICMP, mas não responder a mensagens DHCP. Logo, o recurso ao ICMP serviria apenas para detectar falhas da máquina ou de conectividade, mas não de serviço.

4.3.3.2 Análise de conversações

Esta solução é mais elegante, mas mais complexa.

No instante em que servidor vem acima ambos os sistemas (servidor central DHCP e o iDHCP) respondem a pedidos e qualquer novo pedido é sempre gerado por um cliente.

Este último pode ser de dois tipos:

1 DHCPDISCOVER para um cliente novo na rede
2 DHCPREQUEST
3 DHCPRELEASE
4 DHCPDECLINE
5 DHCINFORM, todos para clientes já configurados

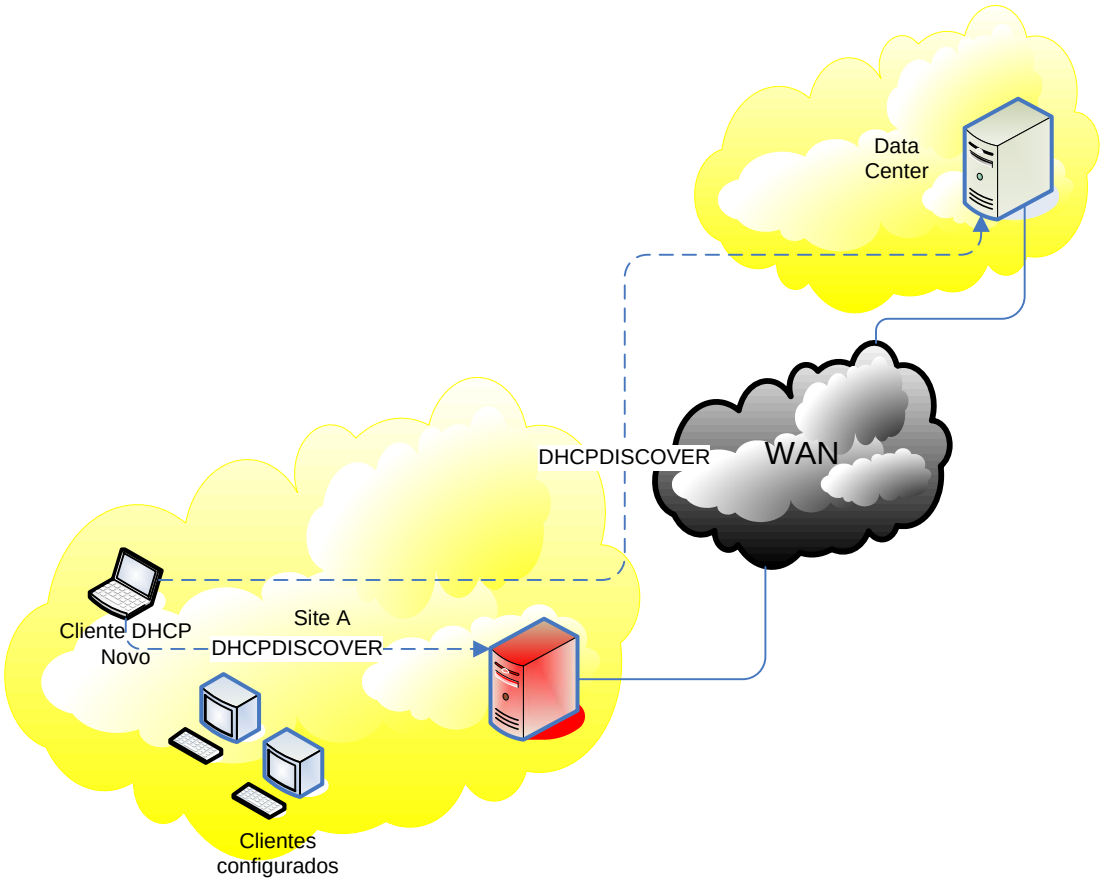

Figura 26: Cliente novo

No primeiro caso (DHCPDISCOVER), a mensagem irá originar uma resposta DHCPOFFER do servidor e uma resposta DHCPOFFER do iDHCP oferecendo os parâmetros de configuração (Figura 26 e 27), isto porque o DHCPDISCOVER se trata de uma mensagem de difusão. Para o cliente, esta situação é equivalente a ter na rede dois servidores DHCP para que possa seleccionar um, situação esta já prevista no protocolo, logo nada de anormal acontecerá.

A captura da mensagem DHCPOFFER do servidor central permite ao iDHCP detectar que o servidor central já se encontra disponível e em consequência disso, deixa de responder a pedidos.

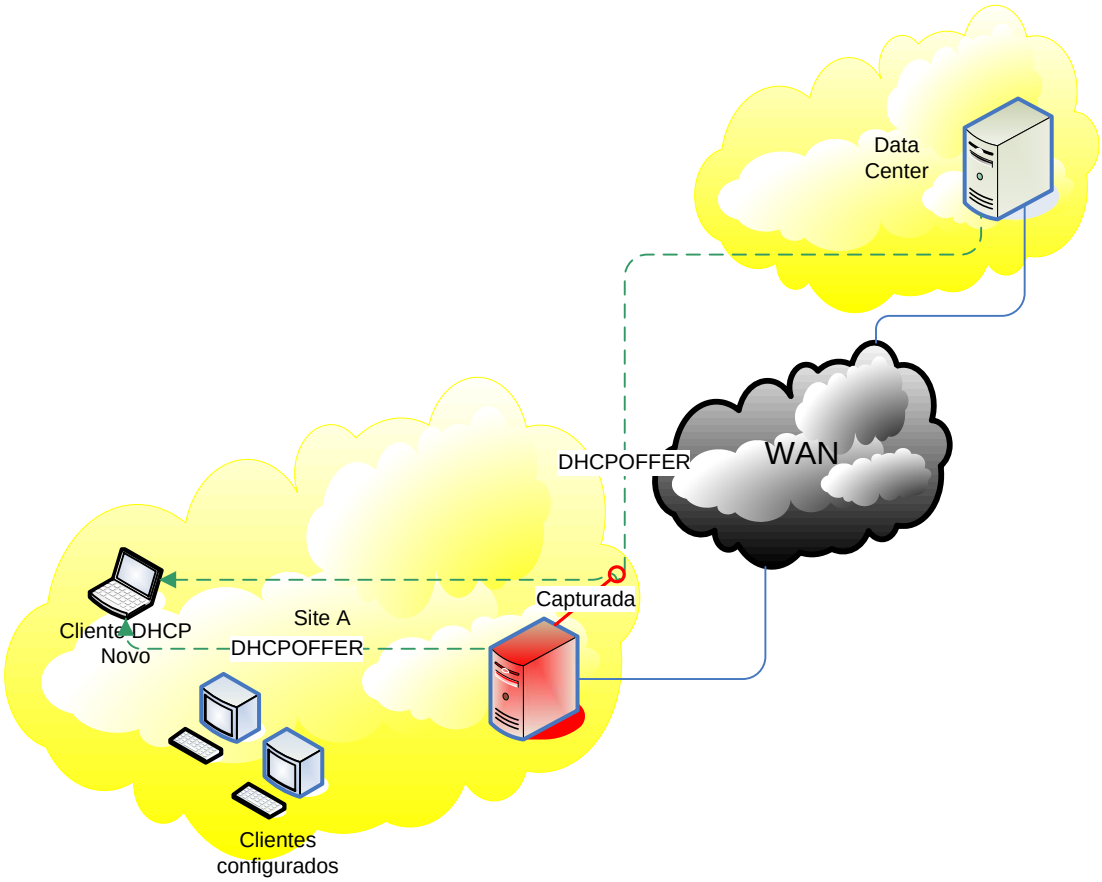

Figura 27: Cliente novo (cont.)

No segundo caso (DHCPREQUEST) temos de distinguir dois casos:

1. O cliente encontra-se no estado RENEWING
2. O cliente encontra-se no estado REBINDING

A distinção existe porque no caso a), a mensagem DHCPREQUEST é uma mensagem unicast, i.e. é apenas direccionada para o servidor que atribuiu a concessão (lease). No caso b), o cliente já desistiu do servidor que lhe atribuiu a concessão (lease) e a mensagem DHCPREQUEST é uma mensagem de difusão (broadcast) para tentar obter respostas de outros servidores.

Se o cliente está no estado **RENEWING**, então a mensagem tem como destino servidor que atribuiu a concessão (lease) que é, por especificação do mecanismo de substituição, sempre o servidor central.

Interessa distinguir dois casos:

- A concessão (lease) foi atribuída antes da falha
- A concessão (lease) foi atribuída depois da falha

No primeiro caso, o servidor irá responder normalmente com uma mensagem DHCPACK de acordo com o protocolo, pois o servidor guarda o estado das concessões que atribuiu até que, por algum motivo, esta seja libertada. O iDHCP ao capturar esta mensagem (DHCPACK) detecta que o servidor central recuperou da indisponibilidade e avança para o estado seguinte. Claro que o iDHCP vai detectar também a mensagem DHCPREQUEST e irá produzir uma mensagem DHCPACK exactamente igual. Uma delas irá ser descartada pelo cliente.

No segundo caso, o iDHCP vai responder ao cliente com uma mensagem DHCPACK (pois tem conhecimento da concessão (lease)) e não vai ter possibilidade de detectar a disponibilidade do servidor pois este não tem o mesmo conhecimento e enviará uma mensagem de erro ao administrador de rede. Neste caso, os dois servidores coexistem, conforme permitido pelo protocolo DHCP.

Se o cliente se encontra no estado **REBINDING (**Figura 28**)**, emite uma mensagem DHCPREQUEST em difusão. Mais uma vez, é importante distinguir quando foi atribuída a concessão (lease):

- A concessão (lease) foi atribuída antes da falha
- A concessão (lease) foi atribuída depois da falha

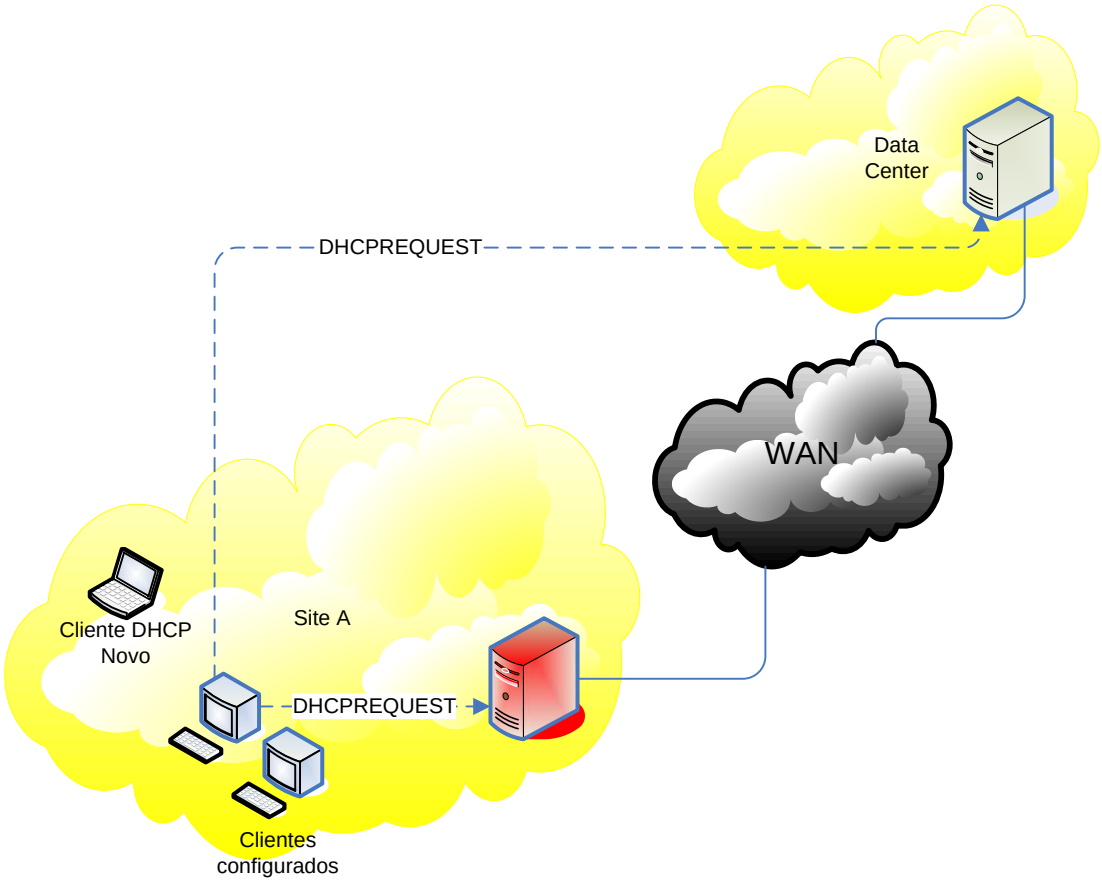

Figura 28: Cliente configurado no estado REBINDING

No primeiro caso, o servidor irá responder normalmente com uma mensagem DHCPACK de acordo com o protocolo. O iDHCP ao capturar esta mensagem (DHCPACK) detecta que o servidor central recuperou da indisponibilidade e avança para o estado seguinte (Figura 29). Antes disso, o iDHCP captura a mensagem DHCPREQUEST e vai reagir de acordo com o seu mecanismo de substituição. Isto irá provocar que o próprio iDHCP envie também ele uma mensagem DHCPACK ao cliente, igual à do servidor.

O protocolo DHCP não prevê que um cliente no estado BOUND receba um DHCPACK. Assumimos que o cliente irá descartar a segunda mensagem DHCPACK, avançando assim o sistema para o estado seguinte, com a recuperação de disponibilidade detectada.

No segundo caso, o servidor não vai responder com um DHCPACK. O iDHCP vai cumprir o seu papel e manterá o cliente activo, mas não conhecerá a recuperação de disponibilidade do servidor. Os dois servidores coexistem até que outra situação aconteça.

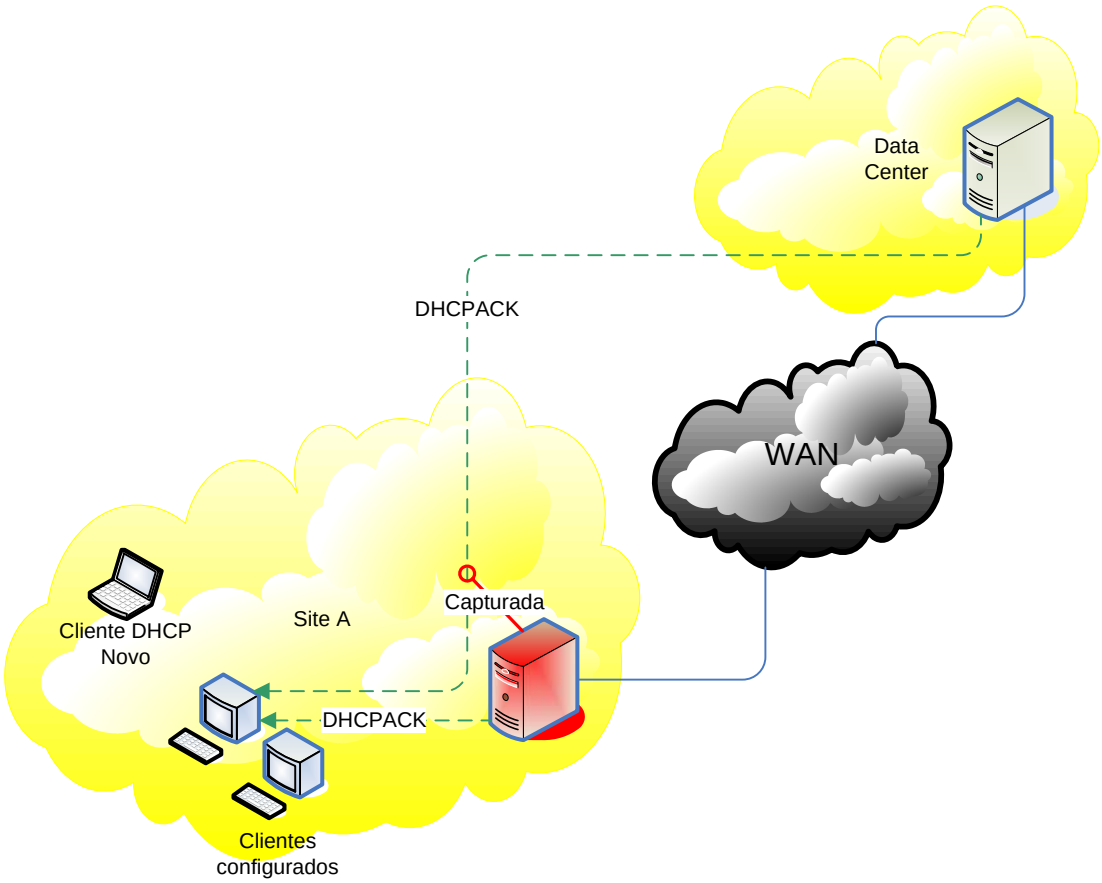

Figura 29: Cliente configurado no estado REBINDING (cont.)

Na terceira mensagem (DHCPRELEASE), temos também de considerar quem atribuiu a concessão (lease) pois esta é sempre uma mensagem com destino definido (unicast).

Se a concessão (lease) foi atribuída pelo servidor central, este ao receber a mensagem DHCPRELEASE, vai proceder internamente de acordo com o protocolo, pois esta mensagem não requer resposta. O mesmo irá ser acontecer se a concessão (lease) tiver sido atribuída pelo iDHCP (ver mecanismo de substituição). Aqui, mais uma vez o iDHCP fica sem poder detectar a recuperação de disponibilidade e coexiste naturalmente com o servidor central, até que outra situação o faça mudar de estado.

No quarto caso (DHCPDECLINE) não é importante quem atribuiu a concessão (lease). Esta mensagem serve apenas para notificar o servidor que o endereço já está a ser utilizado. O servidor de destino limita-se a marcar o endereço como inválido e notifica o administrador do sistema que tal aconteceu. O iDHCP fica sem poder detectar a recuperação de disponibilidade e coexiste naturalmente com o servidor central, até que outra situação o faça mudar de estado.

No último caso (DHCPINFORM), esta mensagem é enviada para o servidor que atribuiu a concessão (lease). Se mais uma vez, tiver sido o iDHCP a fazê-lo, não vai conseguir detectar qualquer resposta por parte do servidor central e coexiste naturalmente com o mesmo até que outra situação se verifique (Figura 30).

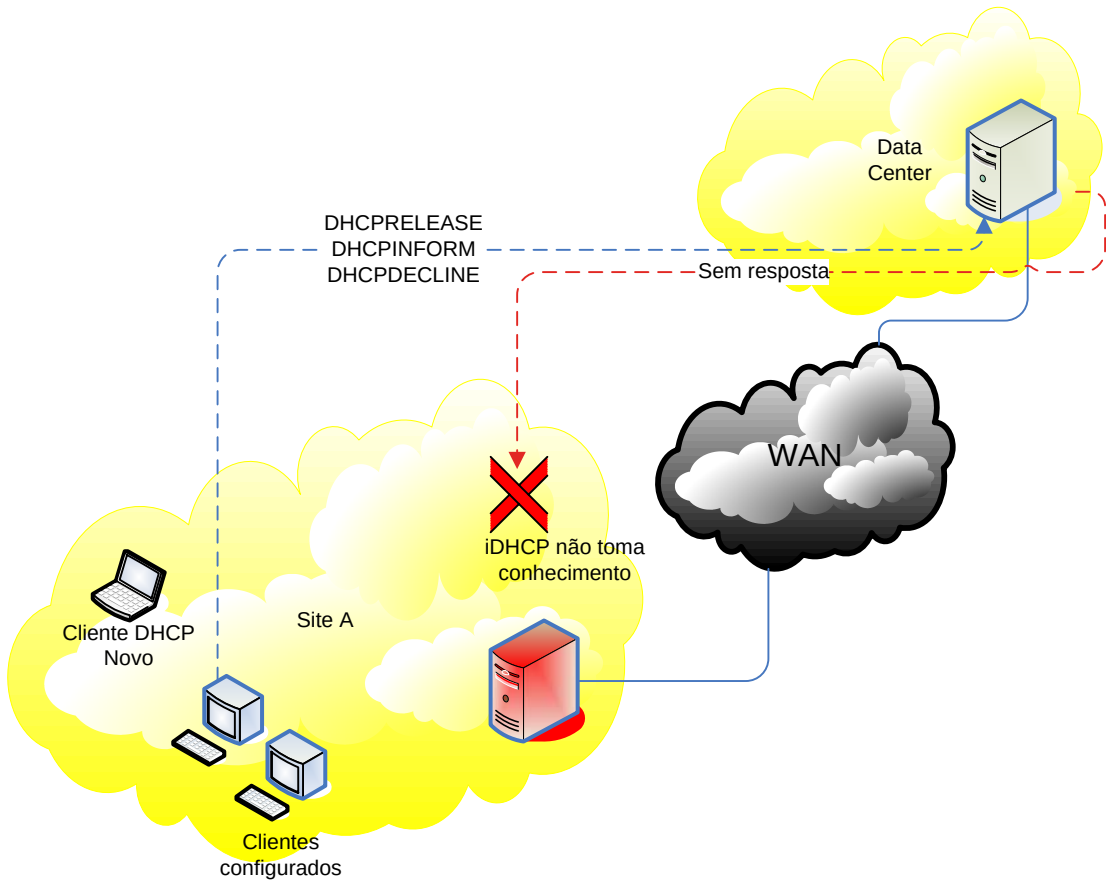

Figura 30: iDHCP desconhece recuperação

Não podemos deixar nenhum DHCPREQUEST sem resposta, pois se tal acontece, corremos o risco de termos um cliente a solicitar um endereço ao servidor que se encontra em baixo, obtendo como consequência uma curta, mas efectiva indisponibilidade de serviço (dependendo da configuração da rede e do serviço) que pode originar até uma desistência, que é o que pretendemos evitar. Na ausência de resposta, responde o iDHCP. Havendo resposta, a mesma serve para detectar a disponibilidade do servidor central.

4.3.4 Actualização do estado do servidor central

Antes de passar para o estado passivo, o iDHCP tem de informar o servidor das alterações que aconteceram na rede, nomeadamente o mapa de endereçamento.

Será mesmo necessário?

Vejamos:

- DHCPDISCOVER

  Mensagem enviada por clientes novos. Se o registo de uma concessão (lease) atribuída pelo iDHCP não constar do repositório de servidor central, este vai entender que se trata da primeira vez que o cliente lhe faz o pedido e irá proceder conforme o protocolo, atribuindo-lhe uma concessão (lease) nova. Do ponto de vista do cliente, só terá o inconveniente (neste caso, um problema menor) de provavelmente, não obter o mesmo endereço que em ocasiões anteriores.

- DHCPREQUEST

  No estado RENEWING, estas mensagens são enviadas para o servidor que atribuiu a concessão (lease).

  No caso de ter sido o servidor central a atribuir a concessão (lease), este ainda terá registo da mesma (o protocolo assim o prevê), e a situação será mais uma situação de normal funcionamento do serviço.

  No caso de ter sido o iDHCP a atribuir a concessão (lease), o servidor central desconhece a existência da mesma.

  Como o mecanismo de substituição define que as mensagens construídas pelo iDHCP levam o endereço IP do servidor central, estas mensagens serão enviadas para o servidor DHCP central (unicast).

  O protocolo não prevê que o servidor DHCP receba mensagens destinadas só a ele (“flag” de broadcast a 0), com informação sobre uma concessão (lease) que desconheça.

Mas o servidor não testa se a mensagem foi enviada a ele ou não, porque o protocolo prevê a existência de vários servidores no mesma gama de endereços a funcionar com um mecanismo que garanta a consistência entre concessões. Ao não testar, só chega à conclusão que recebeu uma mensagem DHCPREQUEST para a qual não tem informação nos seus registos. Logo assume que a mensagem é um DHCPREQUEST que foi enviado de um cliente no estado REBINDING e que outro servidor na rede é que tem autoridade local para estender a concessão (lease). Logo, descarta a mensagem.

Mas como o protocolo não prevê estas situações, isto seria enganar o protocolo.

Além disso, as mensagens que faltam (DHCPDECLINE, DHCPINFORM e DHCPRELEASE) são mensagens que obrigam o servidor a actualizar mapas de endereçamento. Se a informação faltar, o servidor disparará mensagens de erros de configuração para o administrador de rede, o que não é o pretendido. Pretende-se que este sistema seja transparente também para o servidor.

Conclusão: O iDHCP tem de actualizar o mapa de endereçamento do servidor central, perdendo aqui transparência no processo.

4.4 Arquitectura da implementação

Optou-se pelo algoritmo de mais baixo nível para ter um maior controlo sobre os campos e opções do protocolo.

Escolheu-se o PERL como linguagem, devido à sua forte orientação para a programação para redes e por, após pesquisa, possuir os melhores argumentos para ajudar a atingir o objectivo.

A aplicação é composta por um conjunto de subrotinas que desempenham um papel bem definido. Nessas subrotinas usaram-se módulos de PERL já existentes, retirados das livrarias públicas da linguagem, que se adaptavam às necessidades e outros que necessitaram de modificações. Um dos módulos encontrava-se mesmo inacabado e houve necessidade de contribuir para o desenvolvimento do mesmo.

4.4.1 Módulos em PERL

Os módulos em PERL utilizados foram:

- Net::PcapUtils;

Fornece um conjunto de rotinas que simplificam a utilização do módulo Net::Pcap que implementa em Perl uma interface à livraria LBL pcap(3), interface esta que permite uma captura de pacotes da rede ao nível do utilizador (user-level).

- NetPacket::Ethernet;

Fornece um conjunto de rotinas para construir (não implementado) e extrair informação de tramas Ethernet.

- Net::RawSock;

Fornece um conjunto de rotinas que permitem enviar para a rede, pacotes IP previamente construídos.

- NetPacket::IP;

Fornece um conjunto de rotinas que permitem construir (assembling) e extrair informação (disassembling) de pacotes no formato IP.

- NetPacket::UDP;

Fornece um conjunto de rotinas que permitem construir (assembling) e extrair informação (disassembling) de pacotes no formato UDP.

- Net::DHCP::Packet; (modificado)

Fornece um conjunto de métodos que permitem a construção de pacotes DHCP.

Embora seja esta informação prestada pelo autor (Francis van Dun), na realidade o módulo implementa, entre outras, uma rotina de construção de pacotes (‘serialize’) e outra de extracção de informação (‘marshall’).

Nesta última, foi necessário efectuar modificações que permitissem o correcto funcionamento da rotina segundo o protocolo especificado no RFC2131. As modificações efectuadas foram as seguintes:

```
sub marshall {
	use bytes;
	my ($self,$bytes) = @_;
	my $pos = 0;
	$self->{op} = unpack('C',substr($bytes,$pos++,1));
	$self->{htype} = unpack('C',substr($bytes,$pos++,1));
	$self->{hlen} = unpack('C',substr($bytes,$pos++,1));
	$self->{hops} = unpack('C',substr($bytes,$pos++,1));
	$self->{htype} = unpack('C',substr($bytes,$pos++,1));
	$self->{hlen} = unpack('C',substr($bytes,$pos++,1));
	$self->{hops} = unpack('C',substr($bytes,$pos++,1));
	$self->{xid} = substr($bytes,$pos,4); $pos+=4;
	#$self->{xid} = unpack('H',substr($bytes,$pos,4)); $pos+=4; ## jbsoares
	#$self->{secs} = substr($bytes,$pos,2); $pos+=2;
	$self->{secs} = unpack("S",substr($bytes,$pos,2)); $pos+=2; ## jbsoares (1)
	#$self->{flags} = substr($bytes,$pos,2); $pos+=2;
	$self->{flags} = unpack("S",substr($bytes,$pos,2)); $pos+=2; ## jbsoares (2)
	$self->{ciaddr} = inet_ntoa(substr($bytes,$pos,4)); $pos+=4;
	$self->{yiaddr} = inet_ntoa(substr($bytes,$pos,4)); $pos+=4;
	$self->{siaddr} = inet_ntoa(substr($bytes,$pos,4)); $pos+=4;
	$self->{giaddr} = inet_ntoa(substr($bytes,$pos,4)); $pos+=4;
	$self->{chaddr} = mac2str($self-> {hlen},substr($bytes,$pos,16)); $pos+=16;

	$self->{sname} = substr($bytes,$pos,64);$pos+=64;
	$self->{file} = substr($bytes,$pos,128); $pos+=128
	#$self->{options} = unpack("H*",substr($bytes,$pos)); ## jbsoares
	$self->{options} = substr($bytes,$pos); ## jbsoares (3)
	#$self->{options} = new Net::DHCP::Options() -> marshall(substr($bytes,$pos));

	return $self;
}
```

A amarelo encontram-se destacadas as alterações efectivamente adoptadas.
Em (1) e (2) a razão da alteração passa por corrigir a opção do autor ao extrair para os campos 'secs' e 'flags' o respectivo conteúdo binário (o que não é tratável[6]). Segundo o RFC2131 (DHCP), estes dois campos são do tipo "unsigned short", i.e. de comprimento 16 bits sem sinal, logo a extracção é feita seguindo à risca esta informação.

Em (3) a alteração torna-se obrigatória pois o método chamado pertence ao módulo Net::DHCP::Options, sendo este o módulo inacabado tal como foi anteriormente referido. Este módulo não implementa o tratamento das diversas opções presentes nos pacotes, retornando vectores de caracteres nulos.
Assim optou-se por extrair a informação binária como chega, e implementar-se o tratamento das opções que fossem sendo necessárias.

A verde encontra-se uma correcção que se assume como um erro de impressão, pois a versão publicada atribui o campo de 128 bytes novamente ao campo 'sname' do pacote. Foi corrigido para o campo 'file' conforme o RFC2131.

[6] É possível que o autor do módulo tomasse este procedimento numa rotina *a posteriori* colmatando a falha

A cinzento encontram-se destacadas as alterações necessárias para obter uma saída hexadecimal a partir do conteúdo binário (mais facilmente tratável para efeitos de depuração[7]).

4.4.2 Rotina principal

A aplicação não faz mais do que capturar os pacotes de rede de forma contínua (*deamon*) e enviá-los para a rotina principal de tratamento da informação (*process_pkt*). A Figura 31 ilustra isso mesmo.

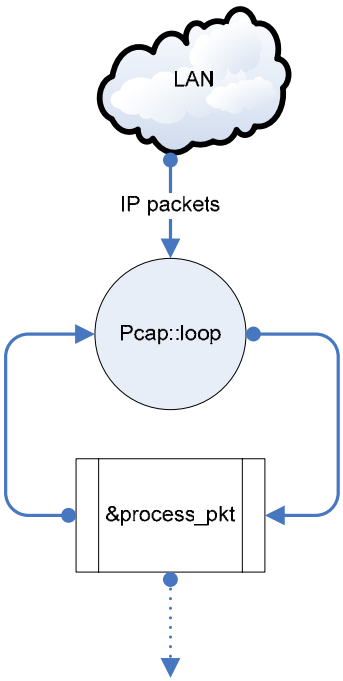

Figura 31: Captura de pacotes

Por sua vez a rotina principal chama sucessivamente as subrotinas de tratamento do pacote aos mais diferentes níveis (ver Figura 32).

[7] A consola reage mal ao imprimir conteúdo binário

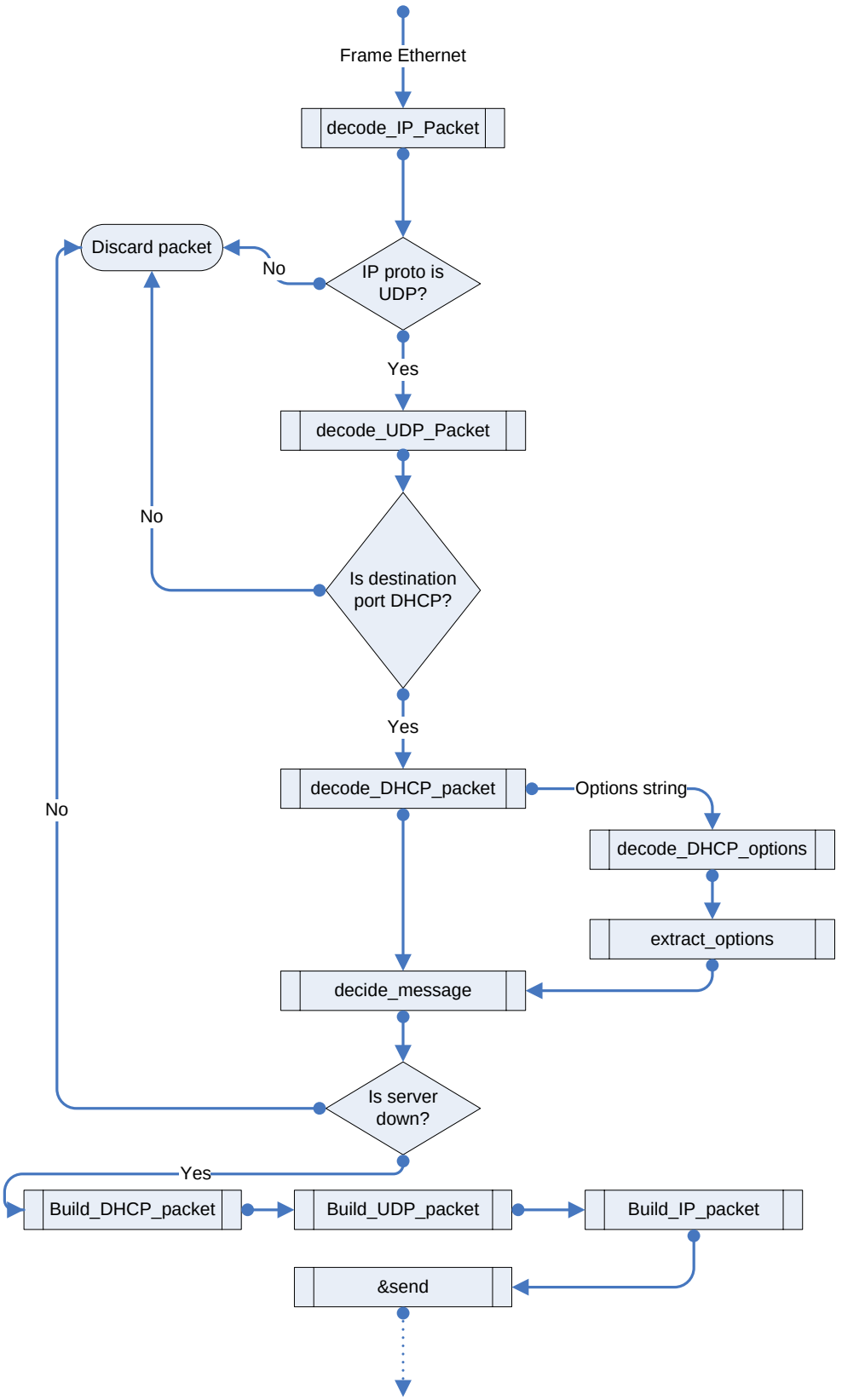

Figura 32: Processamento dos pacotes

4.4.3 Subrotinas

- sub decode_IP_packet{}

Rotina que extrai o pacote IP da trama Ethernet. Todos os campos ficam disponíveis sendo que o campo 'data' contém a informação da camada de transporte (neste caso UDP) em binário. Assim esta informação é passada pela rotina principal à subrotina de descodificação da camada de transporte.

- sub decode_UDP_packet{}

Rotina que extrai o pacote UDP do campo 'data' do pacote IP. O campo 'data' do pacote UDP representa o pacote DHCP em binário.

- sub decode_DHCP_packet{}

Rotina que extrai os campos do pacote DHCP em formato tratável e extrai as opções em binário.

- sub decode_DHCP_options{}

Rotina que determina quais as opções que existem no pacote DHCP em causa. Reproduz-se aqui parte do código desta rotina e da seguinte, pois representam o código em falta nos módulos referidos anteriormente.

```
sub decode_DHCP_options
{
	my ($pkt) = @_;
	my %options_table;
	my $pos = 4;	# MAGIC COOKIE
	my $total = length($pkt);
	#print "Comprimento das Options: ".length($options)."\t Com MAGIC COOKIE";
	while ($pos < $total) {
		my $type = unpack("H2",substr($pkt,$pos,1)); ## Assim vou buscar o hexa
		#print "Nibble: $type";
		last if ($type eq 'ff');# Type 'FF' signals the end. O teste é feito em hexa (nibbles)
		$type = unpack("C",substr($pkt,$pos++,1)); ## Assim vou buscar o valor
		#print "$type";
		my $len = ord(substr($pkt,$pos++,1));
		#print "Comprimento da Opção: $len";
		my $option = substr($pkt,$pos,$len); ## por desempacotar
		$options_table{$type} = $option; ## preenche uma hash table com todas as opções
		$pos+=$len;
		#print "\t\t$pos";
	}
	return %options_table;
}
```

- sub extract_options{}

Rotina que, opção a opção, determina qual o valor atribuído a cada uma. Em baixo vê-se o exemplo para a opção MESSAGE_TYPE, que é a mais importante neste contexto. Cada opção irá chamar a sua própria subrotina (e.g. m_type()) que descodifica o conteúdo da opção de acordo com o formato com que foi construída.

```
sub extract_options
{
	my (%hash) = @_;
	my %options_final;
	foreach(keys %hash){
		#print "$_ : $hash{$_}\n"; ## Imprimo lista inteira
		### Procuro as opções que me interessam e trato-as...

		### DHCP_MESSAGE_TYPE
		if ($_ eq '53'){
			#print "$_:\tDHCP MESSAGE TYPE";
			my $option_value = unpack("C",$hash{$_});
			my $m_type = m_type($option_value);
			#if ($m_type eq 'DHCP_DISCOVER'){reply(DHCP_OFFER)}
			#if ($m_type eq 'DHCP_REQUEST'){ reply(DHCP_ACK)}
			#if ($m_type eq 'DHCP_ACK'){ ignore('Server Side')}
			$options_final{$_} = $m_type;
		}

		}
return %options_final;
}
```

- sub decide_message{}

Rotina que detecta se o servidor central está a funcionar ou não. Em caso de falha, decide qual a mensagem a enviar ao cliente de modo a manter o cliente em cima. Esta é a rotina mais importante do iDHCP.

- sub build_DHCP_packet{}

Rotina que constrói o pacote DHCP adequado, de acordo com a mensagem escolhida pela rotina *decide_message{}*.

- sub build_UDP_packet{}

Rotina que anexa o cabeçalho UDP ao pacote DHCP.

- sub build_IP_packet{}

Rotina que anexa o cabeçalho IP ao pacote UDP.

- sub send{}

Rotina que envia o pacote IP para a rede. Esta rotina é a única deste grupo que necessita de ser chamada por referência, pois existe uma função *built-in* do PERL com o mesmo nome.

4.5 Notas de implementação

4.5.1 Teste de critério de detecção de falha

Ao escolher um critério de detecção de falha foram consideradas duas hipóteses:

1 Aprendizagem dos tempos de resposta existentes na rede e detecção baseada em níveis temporais de fronteira
2 Conhecimento do comportamento do serviço e detecção dos comportamentos esperados dos clientes na ausência do servidor.

No primeiro caso foi sentida uma enorme dificuldade na obtenção de um algoritmo que fosse robusto o suficiente para suportar o carácter imprevisível da rede alargada.

É relativamente fácil medir os tempos de resposta da rede durante um intervalo de tempo e definir critérios de fronteira a partir dos tempos medidos. No entanto, qualquer oscilação na rede que provoque um qualquer tipo de atraso faz com que os níveis de fronteira se tornem inadequados para tomar decisões, correndo-se o risco dessas mesmas decisões serem as erradas (e.g. assumir indisponibilidade quando esta não existe).

Um algoritmo que produzisse resultados satisfatórios, teria de ser um algoritmo continuamente adaptativo, que conhecesse o comportamento da rede em cada instante.

No segundo caso, a complexidade de implementação é muito mais reduzida. Ao optarmos por um critério de contagem obtemos duas coisas significativamente importantes:

1 Imunidade a variações de tempos de resposta da rede
2 Critério de fronteira verdadeiramente determinístico

Por estas razões opta-se pelo segundo caso, pois uma complexidade reduzida de implementação transforma um resultado probabilístico sujeito a falha, num resultado objectivamente determinístico com uma robustez satisfatória.

4.5.2 Algoritmo de aprendizagem

O objectivo a que nos propomos nesta secção passa por decidir qual a melhor estratégia para a implementação do algoritmo de aprendizagem.
As questões que surgem são as seguintes:

- Quais os parâmetros que devem ser alvo de aprendizagem por parte do iDHCP?
- Em que instante consideramos que a aprendizagem foi concluída?

- Com que periodicidade deve o algoritmo efectuar nova aprendizagem (information refresh/update)?

4.5.2.1 Parâmetros alvo

Por análise do protocolo considera-se que o iDHCP tem pleno conhecimento da rede em que foi inserido, quando toma conhecimento de todos os parâmetros que caracterizam a rede em causa e são suficientes para a construção de pacotes DHCP pertencentes aquela rede. São estes:

- Endereço IP do(s) servidor(es) DHCP que servem a rede e vão ser objecto de monitorização
- Gama de endereçamento (scope) que abrange as máquinas da rede
- Tempo de concessão (lease time) na atribuição de endereços pelo servidor

Ao testar o algoritmo, verificou-se que, estando o iDHCP na posse destes parâmetros em conjunto com a informação retirada do pacote em análise no momento, consegue reunir toda a informação necessária para a construção dos pacotes DHCP adequados à rede em causa.

4.5.2.2 Conclusão da aprendizagem

A aprendizagem considera-se concluída quando o iDHCP tiver conhecimento de todos os parâmetros alvo anteriormente mencionados. O instante em que tal acontece depende claramente da configuração do tempo de concessão do endereçamento do lado do servidor, pois é este mesmo o parâmetro que determina a frequência de mensagens cliente-servidor e consequentemente servidor-cliente.

4.5.2.3 Frequência de actualização

A resposta a esta questão depende do cenário considerado. Se assumirmos que o serviço não tem uma configuração estável durante um determinado intervalo de tempo Δt, então a frequência de actualização tem de ser inferior a Δt. Se numa rede alargada com rede de interligação própria (backbone) de uma empresa podíamos assumir uma semana com configuração estável, num ISP podíamos considerar um dia.

Na prática e por simplicidade de implementação, a frequência de actualização é constante utilizando para esse fim todas as mensagens capturadas pelo iDHCP.

4.5.3 Monitorização do estado do servidor

Em todo o seu tempo de funcionamento, o iDHCP tem de ter presente o estado de disponibilidade do servidor central. Para atingir esse fim, foram estudadas duas possibilidades (ver secção 0):

1 Polling
2 Análise de tráfego (conversações DHCP)

A segunda solução possui uma flexibilidade inegável devido ao enorme conjunto de informação disponível. No entanto, nem sempre a opção elegante e complexa é necessariamente a melhor.

Assim verificou-se que nas falhas de conectividade (que são as mais frequentes) a primeira solução (polling) é a mais eficiente, pois detecta rapidamente a recuperação de conectividade com o servidor. Por análise de tráfego a detecção é efectivamente mais lenta além de algoritmicamente mais complexa.

Nas falhas de servidor, o polling verificou-se bastante eficiente quando a falha ocorre no equipamento (hardware). Mas se a falha ocorre no serviço, esta solução toma uma decisão errada se a implementação passar por usar o protocolo ICMP. O teste teria sempre de ser feito ao serviço, só assim o polling possuiria a robustez desejada. O uso do protocolo ICMP implicaria também que o servidor a monitorizar tivesse este serviço activo, o que constituía uma perda desnecessária de autonomia exigida por esta arquitectura, já que não funcionaria bem com servidores que optassem por ter o serviço ICMP desligado.

No entanto, na prática optou-se pela análise de conversações, pois a implementação necessária já é feita para outros fins e pode ser aproveitada para este da mesma forma, mantendo-se também a filosofia de algoritmo inteligente que se pretende para esta aplicação.

4.5.4 Reacção do servidor a um pedido desconhecido

O protocolo já prevê esta possível falha na implementação. O que o servidor faz é ignorar por completo o pedido pois considera que o mesmo não lhe é dirigido especificamente a ele. Esta situação obriga a que o iDHCP actualize o estado do servidor.

A melhor e mais expedita forma de o fazer seria recorrer a um mecanismo de mensagens semelhante ao DHCP Failover Protocol (ver secção 3.2.2), pois contaríamos desde já com a implementação do mecanismo do lado do servidor. Assim aproveitar-se-ia o mecanismo usado para redundância local na redundância distribuída.

5 Conclusões

5.1 O problema

Abordou-se nesta dissertação um problema real que ainda hoje existe associado a serviços centralizados, o problema originado pela perda de conectividade com um servidor remoto gerido centralmente.

Para o estudo do problema, seleccionou-se o serviço DHCP, pois as consequências da dita ausência de conectividade com o servidor remoto tem reflexos nefastos para o normal funcionamento da rede local, i.e. a perda de conectividade com o servidor DHCP remoto, origina em muitos casos, a perda de conectividade dentro da rede local, devido à perda de endereçamento.

Este problema é especialmente sensível no caso de haver mobilidade com novos clientes a ligarem-se à rede a qualquer momento, que neste cenário (ausência de conectividade com o servidor central) ficam totalmente impossibilitados de obter a ligação, de forma automática, à rede local.
Ficam assim obrigados a configurar manualmente a ligação.

Esta configuração manual traz vários inconvenientes. Por um lado, não é habitual que o comum dos utilizadores saiba configurar manualmente a ligação mesmo conhecendo os parâmetros da rede, o que obrigaria a uma linha de suporte permanente para estes casos. Por outro lado, essa mesma configuração manual, ainda que resolvesse o problema de forma imediata, teria de ser desfeita no futuro, quando o normal funcionamento do serviço fosse retomado, pois caso contrário, estas máquinas que sofressem este procedimento, ficariam invisíveis a modificações que viessem a ocorrer centralmente (mais uma vez, a obrigação de uma linha de suporte permanente).

5.2 O serviço DHCP

O serviço DHCP segue o modelo cliente-servidor. As mensagens circulam dos clientes para os servidores na forma de pedidos e circulam dos servidores para os clientes na forma de respostas.

Para implementar a solução proposta, foi necessário conhecer o funcionamento do protocolo em detalhe, com especial ênfase para os comportamentos expectáveis do servidor e do cliente. Foi também necessário conhecer em detalhe o significado de todos os campos e opções que podem surgir num pacote deste serviço.

Conhecer o comportamento expectável do servidor é necessário para que se possa emular um na ausência do central. Conhecer o comportamento expectável do cliente é necessário que se possa construir as mensagens do servidor de forma a poder influenciar o seu comportamento.

O conhecimento do formato do pacote é essencial, pois esta solução propõe-se construir correctamente mensagens do serviço DHCP, i.e. mensagens que obedeçam ao protocolo associado a este serviço.

5.3 A análise

Para verificar a validade da solução, optou-se por efectuar *a priori* uma análise de fiabilidade considerando os vários cenários de utilização do serviço nos dias de hoje.

Consideraram-se os cenários tal como eles são idealmente e após isto consideraram-se as soluções de redundância já adoptadas para lidar com os diversos cenários realisticamente.

Concluiu-se que o problema não é verdadeiramente crítico em cenários de redes locais devido ao curto tempo de actuação do responsável na ocorrência de uma falha.

Demonstrou-se que o problema é verdadeiramente crítico em cenários de utilização deste serviço em redes alargadas de redes locais, geograficamente distribuídas e em cenários de fornecedores de serviços Internet (ISP's).

5.4 A solução

Foi proposta uma solução baseada numa arquitectura de agentes autónomos distribuídos pelas várias redes locais, com um grau de inteligência que permite ao sistema global uma elevada tolerância a falhas de comunicação com o servidor, sejam estas devido a falhas de conectividade como devido a falhas do próprio servidor.

Foi abordada a possível localização destes agentes autónomos, que nos levou a concluir que a sua presença na porta do comutador principal (de acesso, logo após o encaminhador) a funcionar em modo promíscuo, seria perfeitamente transparente à configuração física da rede.

Ficou claro que, a fiabilidade desta solução é superior a todos os cenários de utilização dos dias de hoje, pois resolve de forma explícita os problemas de servidor (extremo do sistema) e de conectividade (entre extremos do sistema).

5.5 Notas sobre segurança

Esta solução só é possível desta forma porque o serviço DHCP é um serviço nitidamente inseguro. Se tal não acontecesse, não seria possível simplesmente capturar o tráfego DHCP e ter acesso a todos os campos e opções de forma legível e tratável.

Esta insegurança do serviço e o aproveitamento da mesma para a implementação desta solução torna a mesma apetecível se puder ser explorada por alguém mal intencionado. Isto constitui uma desvantagem da

solução, pois com toda a facilidade possibilita o aparecimento de pacotes falsos na rede.

Neste momento trabalha-se para a obtenção de um protocolo DHCP seguro em paralelo com o IPv6.

No contexto futuro de um protocolo DHCP seguro, o desenvolvimento aqui feito não seria de todo inválido, bem pelo contrário. A única preocupação neste cenário (que podia ou não ser um constrangimento), era a de incluir um módulo inicial de autenticação no serviço para que se pudesse aceder aos pacotes na sua forma desencriptada.

5.6 Extensão do modelo para outros serviços

Soluções semelhantes poderiam ser adoptadas para serviços que possuam carácter centralizado semelhante ao DHCP.

As vantagens são evidentes. Ainda que desses outros serviços não dependa a possibilidade ou impossibilidade de ligação à rede (é isto o que torna o DHCP tão crítico), a flexibilidade de possuirmos uma gestão centralizada de um serviço sem nos preocuparmos minimamente com a indisponibilidade do mesmo é uma vantagem inegável.

Este mesmo conceito podia mesmo ser adaptado a serviços de carácter não centralizado só pelo facto de dotar esses serviços da facilidade de uma gestão centralizada.

Fica como objecto de estudo para um futuro trabalho, o desenvolvimento de um modelo genérico ou uma plataforma genérica que permita a utilização de uma arquitectura distribuída, mas de gestão centralizada, independentemente do serviço que se escolha para esse fim.

5.7 Conclusão final

Com esta solução é possível, manter um serviço centralizado de tão grande importância para o bom funcionamento de uma rede IP privada como o DHCP, sempre operacional do ponto de vista quer do cliente, quer do servidor.

Estamos em crer que é possível construir uma solução que actue de forma transparente, sem interferir com o normal funcionamento do protocolo e sem abdicar da flexibilidade de gestão que se pretende num serviço centralizado.

Fica para trabalho futuro, a implementação de um verdadeiro mecanismo de aprendizagem, mecanismo este que por si só será com certeza alvo de muita discussão.

No âmbito desta dissertação chegou-se à conclusão da necessidade de uma actualização do estado do servidor após falha. Esta situação impede que a total transparência dos agentes autónomos do ponto de vista do servidor. A estudar fica a hipótese de, através de um mecanismo de aprendizagem verdadeiramente funcional, recorrer às mensagens do protocolo para proceder a essa actualização. O objectivo neste caso seria o emular as mensagens dos clientes ao contrário do que foi âmbito deste trabalho, que foi emular as mensagens do servidor. Outra possibilidade seria esperar pela implementação do DHCP Failover Protocol e tentar usar as mensagens previstas pelo mesmo, como referido anteriormente.

Referências

1. IPv4 Address Space (http://www.iana.org/assignments/ipv4-address-space)
2. Address Allocation for private Internets (RFC1918 - http://www.faqs.org/rfcs/rfc1918.html), Y. Rekhter; B. Moskowitz; D. Karrenberg; G. J. de Groot; E. Lear
3. Dynamic Host Configuration Protocol (RFC2131 - http://www.faqs.org/rfcs/rfc2131.html), R. Droms
4. Redundancy (http://en.wikipedia.org/wiki/Redundancy)
5. Perspective on the Host Requirements RFCs (RFC1127 - http://www.faqs.org/rfcs/rfc1127.html), R. Braden
6. DHCP Options and BOOTP Vendor Extensions (RFC2132 - http://www.faqs.org/rfcs/rfc2132.html), S. Alexander; R. Droms
7. Reliability (http://www.sei.cmu.edu/str/indexes/glossary/reliability.html)
8. DHCP Failover, Cisco Systems Inc. (http://www.cisco.com/univercd/cc/td/doc/product/rtrmgmt/ciscoasu/nr/nr3.0/concepts/cg03.htm)
9. Maximizing Uptime with Redundant DHCP, K. Philpot (http://www.networkcomputing.com/1119/1119ws2.html)

Bibliografia

1. Comprehensive Perl Archive Network (http://www.cpan.org)
2. dhcp.org – Resources for DHCP (http://www.dhcp.org/)
3. FindHosts Tool (http://wormhole.anchorageak.net/FindHosts/)
   Improving on DHCP, M. Minasi (http://www.winnetmag.com/Articles/Index.cfm?ArticleID=2662)
4. Internet Systems Consortium, Inc., DHCP Implementation (http://www.isc.org/index.pl?/sw/dhcp/)
5. Introdução ao PERL, I. Vila Verde (http://paginas.fe.up.pt/~jvv/PERL/perl-intro.html)
6. The Internet Engineering Task Force (http://www.ietf.org/)